

# **AMIGA Frame Grabber**

Helge Böhme

3. November 1997

## **Eigenschaften**

- Digitalisiert einzelne Video-Vollbilder mit  $768 \times 575$  Pixeln in Farbe
- Wird mittels Scart oder Cinch mit jeder beliebigen FBAS-Quelle verbunden
- Arbeitet (theoretisch) Videonorm-unabhängig
- Signalaufbereitung mittels Software im Amiga
- Preview mit 6 Sekunden / Frame auf der Workbench

## **Die Hardware**

- Eine Platine mit  $7 \times 16 \text{cm}^2$  zum Anschluß an den Parallel-Port
- Digitalisiert wird mit dem TDA8708A mit 16MHz
- Gespeichert wird in zwei Video-Frame-DRAMS HM530281R mit je 324KByte
- Netzteil integriert (softwaregesteuert)
- Schaltplan und PCB-Layout als Postscript erhältlich

## **Die Software**

- Geschrieben in AmigaE (von Wouter van Oortmerssen)
- Benutzerinterface in EasyGUI (von Wouter/Jason R. Hulance)
- Preview paßt sich automatisch dem Workbench-Screen an
- Zahlreiche Optionen
- Zur Zeit unterstützte Videonormen: PAL
- Adapiver Dreizeilenkammfilter
- Speichert Rohe Daten, Bildschirm und Echtfarben
- Lokalisiert
- ARexx-Port
- AppIcon und AppWindow
- Rauschfilter und Farbflankenversteilerung
- Unterstützt OCS bis AGA, 68k
- Läuft ab AmigaOS 2.1

## **Der Konstrukteur**

*Helge Böhme  
Ludwigstraße 21  
38106 Braunschweig  
EMail: [h.boehme@tu-bs.de](mailto:h.boehme@tu-bs.de)*

## **Erklärung**

DIESES STÜCK HARD- UND SOFTWARE IST FREI VERTREIBBAR, SOFERN DIE ZUSAMMENSTELLUNG NICHT VERÄNDERT WIRD. DAS KOPIERRECHT LIEGT BEI HELGE BÖHME. BEI NACHBAU BITTET DER AUTOR UM DIE ZUSTELLUNG EINER POSTKARTE (CARDWARE) ODER SONSTIGEN GESCHENKEN (GIFTWARE). BEI AUFNAHME DES PAKETS IN JEGLICHE KÄUFLICHE PRODUKTE IST VORHER SEIN EINVERSTÄNDNIS EINZUHOLEN.

DER AUTOR ÜBERNIMMT KEINERLEI GARANTIE FÜR DIE KORREKTE FUNKTION DER NACHGEBAUTEN HARDWARE, SOWIE DER FEHLERFREIHEIT DER SOFTWARE. IN JEDEM FALL HAF-TET DER AUTOR NICHT FÜR SCHÄDEN BETREFFEND DER SACHE UND DES LEBENS AUFGRUND DIESER ZUSAMMENSTELLUNG.

VORSICHT MIT KOPIERRECHTLICH GESCHÜTZTEN BILDQUELLEN.

# Inhaltsverzeichnis

<b>0</b>	<b>Einleitung – das Konzept</b>	<b>4</b>
<b>1</b>	<b>Die Hardware</b>	<b>5</b>
1.1	Anschlüsse . . . . .	5
1.2	Das Netzteil . . . . .	6
1.3	Der Analogteil . . . . .	6
1.3.1	Synchronisation . . . . .	6
1.3.2	Zur Abtastfrequenz . . . . .	7
1.4	Der Digitalteil . . . . .	8
1.4.1	Die RAMs . . . . .	8
<b>2</b>	<b>Die Software</b>	<b>10</b>
2.1	Schwarzweiß – der einfache Teil . . . . .	10
2.1.1	Die Bildverarbeitungsschritte im einzelnen . . . . .	11
2.2	Digitale Filterung . . . . .	11
2.3	Farbe – der schwierige Teil . . . . .	12
<b>3</b>	<b>Die Bedienungsanleitung</b>	<b>14</b>
3.1	Die Einstellungen . . . . .	15
3.2	Der ARexx-Port . . . . .	18
3.2.1	Die Variablen . . . . .	20
3.3	Was ist noch zu tun? . . . . .	21
3.4	Fehler . . . . .	21
3.4.1	Bekannte Fehler . . . . .	21
	<b>Literatur</b>	<b>22</b>
<b>A</b>	<b>Versionen</b>	<b>22</b>
A.1	Hardware . . . . .	22
A.2	Software . . . . .	23
<b>B</b>	<b>Die Schaltpläne</b>	<b>23</b>
<b>C</b>	<b>Das Platinenlayout</b>	<b>26</b>

D Der Bestückungsplan	28
E Die Bauteileliste	29

## 0 Einleitung – das Konzept

Wie bekomme ich Fernseh- (oder Video-) Bilder in meinen Amiga?

Diese Frage hat mich lange Zeit beschäftigt. Zuhilfe kam mir die Tatsache, daß ich als Student der Elektrotechnik ein paar passende Vorlesungen besuchen durfte:

- Fernsehtechnik I
- Computer-Sehen (Bildverarbeitung und -analyse)
- Fernsehtechnik II

### Echtzeit oder nicht oder wie?

Prinzipiell gibt es drei mögliche Arten von Video-Digitizern:

**Slow-Scan-Digitizer** Digitalisiert einen (oder wenige Pixel) pro Videozeile ( $64\mu s$ ). Dazu muß ein Standbild im RGB-Format vorliegen. Wahrlich keine Echtzeit, ist am besten mit dem Photographieren eines Motivs und anschließenden Einscannen des Bildes zu vergleichen.

**Echtzeit-Digitizer** Diese erledigen den gesamten Signalverarbeitenden Teil mittels Hardware. Auch die Umformatierung der Daten auf Amigaspezifische Formate muß onboard geschehen. Das ganze muß dann mittels DMA ins Chip-Memory geschrieben werden. Eine Aufwendige Hardware für den Prozessor- oder Zorro-Bus.

**Frame-Grabber** Diese sind in der Lage, Videosignale in Echtzeit einzulesen und in Nicht-Echtzeit zu Verarbeiten, d. h. sie liefern Standbilder aus laufenden Videobildern.

Nach Abwägung der Vor- und Nachteile dieser Typen stand das Konzept für meinen Frame-Grabber ...

Die Hardware wird aus Kosten- und Aufwandsgründen möglichst einfach gehalten. Dies hat zudem noch den Vorteil kleiner Abmessungen der Schaltung. Konzipiert wurde sie Schließlich als eine externe Schaltung zum Anschluß an den Parallelport des Amiga (darum kann sie prinzipiell mit nur kleinen Änderungen auch an anderen Plattformen betrieben werden). Die Aufgaben der Hardware beschränken sich auf:

- Auswahl und Verstärkung des Videosignals
- der Synchronisation
- der Digitalisierung
- der Zwischenspeicherung der Daten

- 
- der Übertragung zum Amiga

Die Software hat nun die große Aufgabe die rohen Videosignale zu verarbeiten. Dieses umfaßt im wesentlichen die Filterung, also die Trennung des Helligkeitssignals (der Luminanz) von dem Farbsignal (der Chrominanz) sowie die Farbdekodierung (das sgn. RGB-Splitting) der entsprechenden Norm (PAL, NTSC, SECAM). Die Signalverarbeitung per Software hat nun den großen Vorteil der größeren Flexibilität, der nahezu unbegrenzten Erweiterbarkeit und der größeren Präzision bei der Berechnung. Dazu kommt dann nur noch die Darstellung auf dem Bildschirm sowie das Speichern der Bilder.

## 1 Die Hardware

Möchte man alles verstehen, was hier beschrieben ist, setze ich neben Grundlagen der Elektronik auch die Kenntnis vom Aufbau eines Videosignals voraus. Letzteres ist in der angegebenen Literatur [1],[2],[3],[4],[9] beschrieben. Das Verständnis ist zum Nachbau allerdings nicht erforderlich, nur mit dem Lötkolben sollte man umgehen können (die Platine ist zweiseitig, SMD-Bestückt und hat über 100 Durchkontaktierungen).

### 1.1 Anschlüsse

Abbildung 1: Der Framegrabber im Strapu-Gehäuse

## 1.2 Das Netzteil

Das Netzteil ist konventionell aufgebaut: Kabelklemme, Sicherung, Trafo, Gleichrichter, Kondensator, Spannungsregler (5V), Kondensator. Und doch nicht so konventionell: noch ein Spannungsregler (5V) und noch ein Kondensator. Die zweifache Ausführung deshalb, um die analoge Spannungsversorgung von der digitalen zu entkoppeln. Der Eingangskondensator muß mindestens  $2200\mu\text{F}$  haben, sonst brummt's d. h. des Videosignal ist alle 10ms eingedellt.

Eine zusätzliche Eigenschaft ist die softwaregesteuerte primärseitige Abschaltung der Schaltung. Ein Opto-Triac wird mittels der `BUSY`- (und der 5V-Pullup-)Leitung des Parallelports über einen PNP-Transistor angesteuert. Der verwendete Opto-Triac vom Typ TLP3042 hat einen Zero-Crossing-Detector integriert, so daß beim Schalten keine Spannungssprünge auftreten.

Um den Triac vor Überspannungen zu schützen sind nun noch zwei Schutzelemente vorgesehen: Ein 300V Varistor parallel zu den Eingangsklemmen schützt ihn vor Bursts aus den Netzleitungen (z.B. durch Blitzschlag) und eine Transildiode parallel zu den Trafoklemmen vor dem Ausschaltimpuls der Induktivität des Trafos.

## 1.3 Der Analogteil

Das Herzstück, der Analog-Digital-Wandler TDA8708A, stellt gleichzeitig die Brücke zum Digitalteil dar. Es handelt sich um einen Spezialwandler für Videosignale mit einer maximalen Wandlungsfrequenz von 32MHz. Zur optimalen Aussteuerung hat er einen selbstregelnden Eingangsverstärker (er kann sowohl den Pegel als auch den Offset variieren). Es ist möglich bis zu drei Videoquellen anzuschließen, die sich per Software auswählen lassen. Beschaltet ist dieser gemäß des Datenblattes [7].

Aber der Reihe nach: Das Videosignal von der Scartbuchse oder einer der Cinchbuchsen gelangt nach der obligatorischen Leitungsanpassung und kapazitiver Koppelung an die Eingangspins des TDA. Einer der Pins wird mittels zweier Steuerleitungen ausgewählt und das jeweilige Signal gelangt an den Verstärker. Dort wird es gemäß der Ladungen zweier Kondensatoren (also der Spannung über diese) Verstärkt und mit einem Gleichspannungsoffset versehen. So und nur so wird gewährleistet, daß der A/D-Wandler immer richtig angesteuert wird (schwarz ist schwarz – Wert 64 und weiß ist weiß – Wert ca. 213). Die Ladungen auf den Kondensatoren werden von dem bereits digitalisierten Videosignal abgeleitet, dazu hat der TDA ein paar Komparatoren eingebaut. Das Prinzip ist einfach: Ist das Signal zu hell oder zu dunkel → variere die Ladung auf dem einen Kondensator, hat das Bild zuviel oder zuwenig Kontrast → variere die Ladung auf dem anderen Kondensator.

Nun passiert das Videosignal erst noch einen passiven Filter, um hochfrequente Anteile herauszufiltern damit das Abtasttheorem eingehalten wird (die Abtastfrequenz ist im übrigen 16MHz – warum, dazu später). Vor dem Filter wird das Signal abgegriffen, das für den Syncseparator (LM1881) bestimmt ist. Das digitalisierte Signal gelangt nun auf den Eingangsdatenbus.

### 1.3.1 Synchronisation

Ein endloser Datenstrom nützt wenig, wenn man nicht weiß, wann welche Daten anliegen, um nun alles auf die Bilderfolge zu synchronisieren werden dem analogen Videosignal weitere

Daten entnommen. Der Syncseparator LM1881 [9] entnimmt nun dem Videosignal die Synchronisationsimpulse, die in dem sgn. Bereich „schwärzer als Schwarz“ (digital „0“) vorhanden sind. Um den LM an seinem CVBS-Eingang zu versorgen wird das Videosignal stromverstärkt und mit dem spezifizierten Widerstand von  $75\Omega$  einem Tiefpaß zugeführt. Ganz zufrieden bin ich mit dem LM1881 nicht, da er nicht in der Lage ist aus verrauschtem Signal die Synchronimpulse sauber auszutrennen (es gibt Vergleichstypen der Firma élantec – EL1881, EL1882, EL4581, die sollen besser sein). Der LM gibt folgende digitalen Signale ab:

**CSYNC** Composite Sync, die blanken Synchronimpulse (werden zusammen mit BURST vom TDA benötigt).

**VSYN** Vertical Sync, markiert den Anfang eines Bildes (50Hz-Takt), setzt die Speicher auf den Anfang zurück.

**ODD/EVEN** Zeigt, ob es sich bei dem aktuellen Halbbild um ein gerades oder ungerades handelt, selektiert jeweils einen der beiden Speicher.

**BURST** Markiert die Position des Farbträger-Bursts in der hinteren Schwarzscheule der horizontalen Austastlücke (ist eigentlich nur das Signal eines durch die hintere Flanke von CSYNC getriggerten Monoflops). BURST gelangt wie CSYNC an den TDA (invertiert, und BURST zusätzlich verzögert → zwei einander folgende, sich nicht berührende High-Impulse). Die Signale werden benötigt, um die Verstärkerregelung zu steuern.

### 1.3.2 Zur Abtastfrequenz

Bei der Digitalisierung von Videosignalen gibt es folgende Möglichkeiten zur Wahl des Abtasttaktes:

**Auf die Zeilenfrequenz bezogener Takt** Pro Videozeile (beginnend mit der horizontalen Austastlücke) werden eine feste Zahl von Werten entnommen. Das sind z. B. bei 15625Hz Zeilenfrequenz und 13,5MHz Abtasttakt dann genau 864 Pixel. Damit das immer genau paßt, muß man den Taktgenerator auf das Videosignal synchronisieren → PLL erforderlich.

**Auf den Farbträger bezogener Takt** Es ist auch sinnvoll mit dem vierfachen des Farbträgers von 4,43MHz abzutasten, da sich dann die Farbe sehr einfach demodulieren läßt (auch hierfür ist eine PLL erforderlich). Der Nachteil hierbei ist außerdem die schlechte Anpaßbarkeit auf verschiedene Fernsehnormen (so ist der Farbträger bei NTSC bei 3,58MHz).

**Freilaufender Takt** Der Takt ist völlig unabhängig vom Videosignal. Trotzdem ist es ratsam, ein ganzes Teilverhältnis zur Zeile zu wählen, da das Raster dann annähernd orthogonal ist.

Ich wählte aus Aufwands- und Praktikabilitätsgründen letzteres mit einer Abtastfrequenz von 16MHz, es ergeben sich dann genau 1024 Abtastwerte pro Zeile (ein Videobild hat bei mir also genau 625KByte). Da dieser Takt nicht ganz genau mit dem Videosignal synchron ist, kann es passieren, daß das Bild etwas schräg auf dem Bildschirm erscheint.

## 1.4 Der Digitalteil

Gesteuert wird die ganze Schaltung durch den Parallelport (es gibt also keine Schalter und Tasten zur Betätigung). Zunächst wird die SEL-Leitung dazu benutzt, um von der Amiga-Seite aus die Datenrichtung festzulegen (SEL=„1“: Amiga → Framegrabber, SEL=„0“ umgekehrt). Empfangene Daten landen dabei immer in den Eingangsregistern (IC9, IC10), gesendet werden die Daten direkt von den RAMs. Es wird das Standard-Handshakeverfahren der Parallelschnittstelle verwendet, so werden die Daten mit dem  $\overline{\text{STROBE}}$ -Impuls übergeben bzw. angefordert. Der  $\overline{\text{ACK}}$ -Impuls wird direkt aus  $\overline{\text{STROBE}}$  gewonnen, indem er invertiert wird – die Schaltung arbeitet schnell genug, um die Daten zu senden/empfangen. Ich möchte hierbei darauf hinweisen, daß es unbedingt erforderlich ist, eine Leitungsanpassung für  $\overline{\text{STROBE}}$  zu schaffen und  $\overline{\text{ACK}}$  nicht direkt mit den RCK-Eingängen der RAMs zu verbinden. Normalerweise wird der Framegrabber mit einem Kabel am Port angeschlossen, dort gibt es dann Laufzeiten und Reflexionen der Signale, das führt dann zum mehrmaligen Triggern der RAMs und es gehen Daten verloren.

Was wird nun in den Registern gespeichert:

Bit	Name	Aufgabe
D0	ReadReset	setzt die Leseadresse der RAMs zurück
D1	ReadOdd/-Even	wählt das RAM, aus dem gelesen werden soll
D2	I0	legen den Videoeingangskanal des TDA8708A fest
D3	I1	
D4	Freeze	legt fest, ob neue digitalisierte Daten in die RAMs geschrieben werden sollen

Tabelle 1: Die Registerbelegung des Framegrabbers

Das Signal  $\overline{\text{Freeze}}$  wird durch ein weiteres FlipFlop geführt, welches von dem ODD/EVEN-Signal getriggert wird. Dies gewährleistet, daß das Bild nicht mittendrin eingefroren wird.

Es bleibt noch zu erwähnen, daß es einen einfachen Resetgenerator (bestehend aus einer R-C-Kombination und einem Inverter) gibt, der die FFs und die RAMs nach dem Einschalten definiert zurücksetzt.

### 1.4.1 Die RAMs

Bei den HM530281Rs [8] handelt es sich um 331776-word×8-bit Frame DRAMs. Es sind speziell für digitale Videoanwendungen konzipierte Chips. Sie haben zwei voneinander vollkommen unabhängige Datenports (einen zum schreiben, einen zum lesen), zwei interne Adreßzähler (es gibt also keinen Adreßbus) und einen integrierten Refreshcontroller. Man braucht sich also um nichts weiteres zu kümmern, als um die Daten, die rein und raus sollen, die entsprechenden Schreib- und Lesetakte und die Resetimpulse, die die internen Zähler zurücksetzen. Das klingt gut und funktioniert auch wunderbar.

Ich möchte aber doch auf den Nachteil dieser Chips hinweisen: sie sind extrem schwierig zu besorgen, bei den üblichen (Versand-)Händlern bekommt man sie nicht. Also verweise ich im



allgemeinen auf die Hitachi-Website, wo man eine Liste der Distributoren erhält, die Hiatchi führen, und im speziellen auf den einen Distributor, der sie mir verschafft hat (zu 40DM/Stück):

*CED Ditronic GmbH*  
*Julius-Hölder-Str.42, D-70597 Stuttgart*  
*Postfach 700159, D-70571 Stuttgart*  
*Tel.: +49 (0)711/720001-0, Fax.: +49 (0)711/7289780*

*Vertriebsniederlassung Hannover*  
*– Herrn Aschenberg –*  
*Karl Wichert Allee 66, D-30625 Hannover*  
*Tel.: +49 (0)511/54269-18, Fax.: +49 (0)511/64269-20*

Die Firma *Neumüller Fenner Elektronik GmbH* hat sich um die Chips bemüht (zu 17,60DM/Stück), es gelang ihr allerdings nicht sie zu besorgen, die Firma *MSC Vertriebs GmbH* räumte sich eine Lieferzeit von 7 Monaten(!) ein.

Weitere (europäische) Distributoren findet man hier:

<http://www.hitachi-eu.com/hel/ecg/dist.htm>

Abbildung 2: Der Framegrabber, geöffnet (die Änderungen sind mittlerweile in das Layout integriert)

## 2 Die Software

Bei der Beschreibung der Software möchte ich mich auf die Erklärung der Signalverarbeitenden Komponenten beschränken.

### 2.1 Schwarzweiß – der einfache Teil

Es gibt zwei Anzeigemodi:

1. die rohe Ausgabe der Daten (sie werden genau wie sie von der Hardware kommen dargestellt – evtl. gedithert auf die vorhandenen Grauwerte),
2. die auf den gültigen Bildbereich beschränkte, auf 4:3 skalierte, schwarzwertgeklemmte, kontrastgeregelte und evtl. gefilterte Ausgabe.

Die rohe Ausgabe zeigt dabei auch die Informationen des Fernsehbildes, die sonst unsichtbar sind. Dies sind im einzelnen:

- die Synchronimpulse (der Bereich „schwärzer als Schwarz“) inklusive der sgn. Vor- und Nachtrabanten der vertikalen Austastlücke
- die vordere und hintere Schwarzschulter der horizontalen Austastlücke – dieser Bereich ist jetzt nicht schwarz sondern dunkelgrau.
- der Farbträgerburst in der hinteren Schwarzschulter
- Zusatzsignale in der vertikalen Austastlücke (Videotext, VPS und ein paar Meßsignale)

Abbildung 3: Die Austastlücke

### 2.1.1 Die Bildverarbeitungsschritte im einzelnen

Zunächst besteht das Bild aus zwei Halbbildern, die nacheinander eingelesen und auch nacheinander verarbeitet werden:

**Zentrierung** oder auch Synchronisation; der Bildausschnitt wird richtig positioniert, die  $896 \times 575$  Pixel des Bildes werden aus den  $1024 \times 625$  Pixeln ausgeschnitten.

**Filterung** Handelt es sich um ein Farbbild, kann man mittels digitaler Filter den Farbträger entfernen (siehe dazu das nächste Kapitel).

**Skalierung** Das Bild wird auf 4:3 ( $768 \times 575$ ) entzerrt.

**Schwarzwertklemmung** Der hinteren Schwarzscharakter wird der Pegel für Schwarz entnommen (in der Regel 64), dieser wird von den sichtbaren Pixeln abgezogen.

**Kontrastregelung** Der Schwarzwert entspricht 300mV (bezogen auf die Synchronsignale) des Videosignals. Weiß entspricht dann genau 1V, der Digitalwert für Weiß wird aus dem Schwarzwert und diesem Verhältnis gebildet (in der Regel 213). Die Pixel werden dementsprechend auf den Bereich von 0 bis 255 umgerechnet.

**Rauschfilterung** Optional wird ein verrauschtes Bild mittels eines IIR-Tiefpasses geglättet. Damit das Bild dabei nicht zu stark verwischt, wird die Filterung bei Erkennung von Konturen ausgesetzt. (Diese Vorgehensweise hat sich als effektiver und einfacher herausgestellt, als ein Median-Filter.)

## 2.2 Digitale Filterung

Es gibt zwei Möglichkeiten, den Farbträger ( $4433593,75\text{Hz}$  bei PAL – das entspricht einer Periode von  $\sim 3,6$  Pixeln) aus dem Signal herauszufiltern:

- Eindimensional – der übliche Weg bei einfachen Fernsehgeräten
- Zweidimensional – dieser Weg ist nur mittels digitaler Signalverarbeitung mit vernünftigen Aufwand zu realisieren

Eindimensionale Filterung im digitalen Videobereich wird in der Regel mittels der FIR<sup>1</sup>-Filter realisiert [4]. Dabei werden in einer Zeile mehrere benachbarte Pixel mit einem Faktor versehen und addiert. Die Anzahl der zugrundegelegten Pixel bestimmt die Ordnung des Filters.

Beispiel: Einen Tiefpaß realisiert man dabei mit den für alle  $n$  Pixel gleichen Faktor  $\frac{1}{n}$  (das entspricht der Durchschnittsbildung)

Zweidimensionale Filter werden auch Kammfilter genannt. Der Name Kammfilter kommt von dem kammförmigen Spektrum, das genau dem Spektrum des Farbträgers angepaßt ist [1]. Ein zweidimensionaler Filter berücksichtigt auch die Pixel der benachbarten Zeilen (so ist

---

<sup>1</sup>finite impulse response

der Farbträger bei PAL gegenüber der vorletzten und der übernächsten Zeile genau um  $180^\circ$  verschoben). Da die Zeilen dafür gespeichert werden müssen, erklärt sich damit die Notwendigkeit der digitalen Signalverarbeitung. Treten nun Farb- oder Helligkeitssprünge von Zeile zu Zeile auf, so werden diese jetzt verwischt, dieses zu erkennen und auszugleichen ist die Eigenschaft der sgn. adaptiven Kammfilter, welche von drei Zeilen die nicht berücksichtigt, die vom Mittelwert allzu stark abweicht.

Abbildung 4: Das Testbild, zweidimensional gefiltert

Das Testbild (die `.video`-Datei dafür wurde mittels des Programms `ZonePlateStream(.e)` erzeugt) enthält neben den obligatorischen Grauwert- und Farbbalken die sgn. Zonenplatte. Die Zonenplatte hat nun die Eigenschaft, alle in einem Videosignal möglichen Frequenzen zu enthalten, die derart angeordnet sind, daß die 2d-Fouriertransformierte wieder eine Zonenplatte ergibt. Abb. 4 zeigt, welche Frequenzen der Kammfilter entfernt. Der Vorteil ist hierbei, daß senkrechte hochfrequente Strukturen erhalten bleiben.

Es bleibt noch zu erwähnen, daß das Schwarzweißbild (die Luminaz) sich aus der Differenz von dem rohen Videosignal und dem herausgefilterten Farbsignal (der Chrominanz) ergibt.

### 2.3 Farbe – der schwierige Teil

Um es vorwegzunehmen, die Qualität der Farbberechnung hängt zu einem Großteil von der Qualität der Filter ab. Ist der Filter nicht perfekt, werden Luminaz und Chrominaz nicht sauber voneinander getrennt und es kommt zu Übersprecheffekten. Abgesehen davon gliedert sich die Farbdekodierung wie folgt:

**Rückgewinnung der Farbträgerphase** Das Modulationsverfahren ist die Quadraturamplitudenmodulation, das Signal enthält *zwei* Komponenten (diese sind bei PAL  $U$  und  $V$ ), die um  $90^\circ$  versetzt sind. Zur Demodulation ist die Kenntnis über die Phase nötig. Dazu wird in der horizontalen Austastlücke eine Referenz gesendet (der Farbträgerburst – ca. 10 Schwingungen mit  $135^\circ$ ), mit der sich der (digitale) Oszillator synchronisieren läßt. Praktisch wird dazu der Burst normal demoduliert (s.u.) und dann der Phasenfehler ausgeglichen.

**Synchrodemodulation** Die Farbkomponenten  $U$  und  $V$  werden aus der Chrominanz  $C$  errechnet [1]:

$$\begin{aligned} U &= s \cdot C \cdot \cos \beta \\ V &= s \cdot (\pm) C \cdot \sin \beta \end{aligned}$$

$s$  ist dabei der Faktor, der die Farbsättigung angibt. Er errechnet sich aus der Amplitude des Farbträgerbursts, dieser ist zu 75% gesättigt und hat in der Regel eine Amplitude von 20% des Weißwerts (gegebener Schwarz), also etwa 140mV. Daß die Phase von  $V$  von Zeile zu Zeile umgekehrt wird, ist PAL-Eigenschaft und wird unten erläutert.  $\beta$  ist nun der Wert des digitalen Oszillator-Inkrementes. Praktisch realisiert wurde dieses mit einer Sinustabelle mit 4096 Einträgen. Das Inkrement ist bei einer Abtastfrequenz von 16MHz dann genau 1135.

**Tiefpaßfilterung der Farbkomponenten**  $U$  und  $V$  sind jetzt noch mit dem  $\sin^2$ -förmigen Farbträger versehen, dieser wird mittels eines (FIR-) Tiefpaßfilters entfernt.

**PAL-Kompensation** Um mögliche (prinzipbedingte) Phasenfehler (und somit Farbverfälschungen) ausgleichen zu können, ist die Phase der  $V$ -Komponente Zeilenweise um  $180^\circ$  verschoben. Hebt man diese Phasenverschiebung beim Dekodieren auf (invertieren jeder zweiten Zeile) und mittelt die  $U$ - und  $V$ -Komponenten Zeilenweise, so kompensieren sich gleiche Phasenfehler in den Zeilen. Da dadurch die Farbauflösung in der Vertikalen halbiert wird, kann man einen Schwellwert setzen, über dem dieser Mechanismus abgeschaltet wird.

**Farbflankenversteilerung** (Optional) Da das Farbsignal prinzipbedingt eine geringere Bandbreite hat, verwischen starke Farbkontraste. Diese können erkannt und verstärkt werden, indem benachbarte Pixel verschoben werden. Wählt man den Faktor der Verschiebung zu groß, werden allerdings auch Farbflanken verstärkt, die von vornherein nicht steil waren (z.B. in unscharfen Bildbereichen).

**Dematrizierung** Aus der Luminanz  $Y$  und den Komponenten  $U$ ,  $V$  errechnen sich  $R, G, B$  wie folgt:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1,14 \\ 1 & -0,37 & -0,57 \\ 1 & 2,04 & 0 \end{pmatrix} \cdot \begin{pmatrix} Y \\ U \\ V \end{pmatrix}$$

### 3 Die Bedienungsanleitung

Nach dem Start des Programms „FrameGrab“ wird die Hardware eingeschaltet. Reagiert diese nicht, wird das erkannt, es können immer aber noch Dateien geladen werden. Läuft bereits ein „FrameGrab“, so stehen dem zweiten Programm weder die Hardware noch der ARexx-Port zur Verfügung. Das Programm kann von der Workbench oder aus der Kommandozeile gestartet werden:

```
FrameGrab FILE,ICONIFIED/S,NOMESSAGES/S,INTERRUPT/S,PUBSCREEN/K
```

FILE Dateiname einer zu ladenden „.video“-Datei

ICONIFIED Startet das Programm als AppIcon

NOMESSAGES Unterbindet einen Großteil der Warnrequester

Die beiden letzten Parameter erleichtern den Start aus einem ARexx-Script heraus.

INTERRUPT Es wird zur Paralelportansteuerung eine Interruptroutine installiert. Diese ist langsamer, aber evtl. sicherer.

PUBSCREEN Gibt den Namen des Public-Screens an, auf dem das Benutzerinterface geöffnet werden soll (normalerweise der Default-PubScreen).

Das Programm präsentiert sich mit der Gadtools-Basierten Oberfläche EasyGUI<sup>2</sup>:

Abbildung 5: Das FrameGrab Hauptfenster

Im Hauptfenster befinden sich der Preview-Bereich, der das laufende Videobild mit verminderter Qualität anzeigt (ein Refresh dauert auf meinem A4000LC040 immerhin ca. 6 Sekunden)

---

<sup>2</sup>von Wouter van Oortmerssen und Jason R. Hulance

sowie ein paar Action-Buttons. Diese sind fast alle „sticky“ d. h. sie bleiben nach Anwahl eingedrückt und zeigen somit auch den Status des Programms an. Die Funktionen der Buttons im einzelnen:

**Oeffnen** (o) Öffnet (oder schließt) eine Videodatei zur Anzeige und Bildverarbeitung. Eine geöffnete Datei verhält sich genau so, wie ein Bild aus dem Framegrabber (abgesehen von der Synchronisation).

**Speichern** (s) Speichert die aktuellen Daten (vom Grabber oder aus einer Datei). Es stehen drei Möglichkeiten zur Auswahl:

1. Der rohe Videodatenstrom: das Videobild wird 1:1 in eine Datei geschrieben (625KByte), die später mit „Open“ wieder eingeladen werden kann.
2. Render-Screen: Der Render-Screen wird genau so, wie er ist in eine IFF-ILBM-Datei geschrieben.
3. True-Color: Es werden sämtliche (ausgewählten) Bildverarbeitungsschritte durchgeführt und die 24-Bit-Daten als IFF-ILBM24-Datei gespeichert. Um ein berechnetes Halbbild dabei zwischenspeichern, wird etwas RAM angefordert (max. 961536 Bytes bei Farbe mit Synchronimpulsen).

**Optionen** (p) Das Einstellungsfenster wird geöffnet (oder geschlossen).

**Halt** (f) Ist die Hardware aktiv, so kann man das laufende Videobild einfrieren (wird beim Speichern oder Rendern automatisch aktiv).

**Rendern** (r) Startet alle (ausgewählten) Bildverarbeitungsschritte und stellt das Bild auf dem bestmöglichen Screenmode dar.

**Ende** (q) No Comment.

Es gibt dann noch ein kleines Menü:

**Über** Ein kleiner Informationsrequester öffnet sich.

**Ikoniifizieren** Das Fenster wird geschlossen und durch ein AppIcon ersetzt. Man kann „.video“-Dateien laden, indem man deren Icons auf das Icon (oder auch auf das Hauptfenster) fallenläßt (diese Vorgehensweise dürfte aber wohl bekannt sein).

**Ende** s.o.

## 3.1 Die Einstellungen

Das Einstellungsfenster gliedert sich in sechs Untergruppen, die untere Buttonreihe ist dabei immer präsent:

**Speichern** Die Einstellungen werden dauerhaft in „ENVARC:GrabPrefs“ gespeichert, übernommen und das Einstellungsfenster geschlossen.

**Benutzen** Die Einstellungen werden temporär in „ENV:GrabPrefs“ gespeichert, übernommen und das Fenster geschlossen.

**Zurück** Alle Änderungen seit dem Öffnen des Fensters werden rückgängig gemacht.

**Abbruch** Wie Zurück, das Fenster wird nun aber geschlossen (das Close-Gadget hat die gleiche Wirkung).

Abbildung 6: Das Allgemein-Einstellungsfenster

**Wähle Grafikmodus** Ein Screenmoderequester wird geöffnet, der Bildschirmmodus für den Renderscreen wird ausgewählt und angezeigt.

**Benutze HAM** Öffne einen HAM-Screen zum Rendern, falls erforderlich.

**Benutze FS-Dither** Benutze zum Rendern das Floyd-Steinberg Dithering.

**Fenster Befestigen** Speichert die augenblickliche Position der Fenster als Standard.

Abbildung 7: Das Eingabe-Einstellungsfenster

**Eingabekanal** Wählt einen der drei Video-Eingabeports (Scart, Cinch1 oder Cinch2).

**Dateien Puffern** Dateien werden im RAM zwischengespeichert, das erhöht die Geschwindigkeit, benötigt aber weitere 625KByte RAM.

**Synchronisation per Software** Es wird versucht, nicht vollständig übereinanderliegende Halbbilder (verursacht durch verrauschte Videosignale) Zeilenweise zu synchronisieren.



Abbildung 8: Das Ausgabe-Einstellungsfenster

**Speichern als** legt den Speichermodus fest (Video-Frame, Render-Screen, True-Color oder Fragen).

**Dateiendungen** aktiviert das automatische Anhängen der passenden Endungen („`.video`“, „`.ilbm`“, „`.ilbm24`“) an die Dateinamen (kann im Save-Requester immer noch geändert werden).

**Dateien numerieren** dto. für die Nummerierung der Dateien.

Abbildung 9: Das Vorschau-Einstellungsfenster

**Floyd-Steinberg-Dither** für das Vorschaufenster,

**Glätten** für das Vorschaufenster, aktiviert die Tiefpaßfilterung,

**Halbbild** wählt das gerade oder das ungerade Halbbild für das Vorschaufenster.

Abbildung 10: Das Luminanz-Einstellungsfenster

**zeige Synchronimpulse** aktiviert die Anzeige und das Speichern auch der Synchronimpulse und der Austastlücken samt Zusatzsignale. Deaktiviert den Farbträgerfilter im Schwarz-weißmodus.

**Farbträger filtern** Schaltet die Farbträgerfalle für die Luminanz an.

**Kammfilter** aktiviert den Kammfilter (0 ist aus), der Schwellwert für die Adaptivität läßt sich einstellen (ein guter Wert ist 4).

**Rauschunterdrückung** erlaubt die Aktivierung und Einstellung der Intensität der Rauschunterdrückung (IIR-Tiefpaß). Wirkt auch auf die Chrominanz.

**Bildsignal** Legt den Pegel des Videosignals fest (hängt vom Sender ab – normal sind 700mV gegenüber Schwarz), wirkt wie ein Kontrastregler.

**Ber** echne ermittelt aus den Bilddaten den Maximalwert und stellt den Bildsignalpegel entsprechend ein.

Abbildung 11: Das Chrominanz-Einstellungsfenster

**Farbdekoder** wählt den Farbdekoder. Gegenwärtig nur „keiner“ (Schwarz-Weiß) und PAL („Vierquadrantenmultiplizierer“ wie in Kap. 2.3).

**PAL-Kompensation** aktiviert die zeilenweise Mittelung der Chrominanz und legt den Schwellwert dafür fest (gute Schwellwerte sind zwischen 4 und 6).

**Chroma-Pegel** legt schließlich die bekannte Farbsättigung fest (gute Werte sind zwischen 40 und 50).

**Farbflankenversteilerung** aktiviert schließlich dieselbe. Werte größer als zwei sind mit Vorsicht zu verwenden, da auch unscharf gewünschte Farbflanken verstärkt werden.

## 3.2 Der ARexx-Port

Beim Programmstart wird ein ARexx-Port namens „FGRAB“ eingerichtet. Folgende Kommandos werden unterstützt (in [] optional, in <> gefordert):

**FREEZE** Schaltet zwischen laufendem Videobild und Standbild um. Hat nur bei aktivierter Hardware (keine Datei geladen) Wirkung.

RENDER Das aktuelle Videobild wird gezeichnet.

QUIT ...

ICON (De-)Ikonifiziert.

SAVE [file] Speichert das Bild im aktuellen Modus. Ein Neuer Dateiname kann übergeben werden.

OPEN [file] Öffnet die angegebene (oder letztgenannte) „.video“-Datei.

CLOSE Die (geöffnete) Datei wird geschlossen und wieder auf den Parallelport umgeschaltet.

CALCPICLEVEL Berechnet den Bildsignalpegel neu (entspricht genau dem Button in den Luminanz-Einstellungen).

SET <var> <value> Setzt die angegebene Variable auf den Wert.

QUERY <var> Liest die angegebene Variable aus und übergibt den Wert der 'RESULT'-Variable von ARexx.

### Ein Beispielscript

```
/* Framegrabber Beispielscript
   Aufnahme einer Bildsequenz */
options results
parse arg name increment          /* Kommandozeile: FILE INCREMENT */
if increment <=0 then increment=5
if name==' ' then name='T:Test1'
address command
if (show('P','FGRAB'))==0 then do
    'run FrameGrab iconified nomessages' /* FrameGrab starten, falls es noch */
    'wait 1 secs'                        /* nicht läuft */
end
hours=time('H')
minutes=time('M')-60*hours        /* Zeit ermitteln */
minutes=trunc(minutes/increment)*increment
address fgrab
'SET DISPMESSAGES FALSE'
'SET FILENUM TRUE'
'QUERY SAVEAS'
if result='ASK' then 'SET SAVEAS TRUECOLOR'
'SET FILENAME ' || name           /* Variablen von FrameGrab setzen */
address command wait 1 secs
do forever
    minutes=minutes+increment      /* Endzeitpunkt ermitteln */
    if minutes>=60 then do
        minutes=minutes-60
        hours=hours+1
        if hours==24 then hours=0
```

## 3.2.1 Die Variablen

Variable	Werte	Bemerkung
MODEID	integer	der rohe Grafikmodus
HAM	TRUE FALSE	
FLOYD	TRUE FALSE	
INPCHANNEL	SCART CINCH1 CINCH2	
FILEBUF	TRUE FALSE	
SOFTSYNC	TRUE FALSE	
SAVEAS	ASK VIDEO SCREEN TRUECOLOR	
FILEEXT	TRUE FALSE	
FILENUM	TRUE FALSE	
PREVIEWFS	TRUE FALSE	
PREVIEWSM	TRUE FALSE	
PREVIEWFR	EVEN ODD	
SYNCS	TRUE FALSE	
CHRFILTER	TRUE FALSE	
COMBFILTER	integer[0-10]	
NOISERED	integer[0-4]	
PICLEVEL	integer[50-90]	
CHROMADEC	NONE PAL	
PALCOMP	integer[0-10]	
CHROMALEVEL	integer[0-100]	
COLEDGEENH	integer[0-4]	
BUSY	TRUE FALSE	READONLY Preview wird gezeichnet
FROZEN	TRUE FALSE	
ICONIFIED	TRUE FALSE	
DISPMESSAGES	TRUE FALSE	
FILENAME	string	
		entspr. 500mV bis 900mV
		READONLY Bild eingefrohren
		READONLY
		Warnings ausgeben

Tabelle 2: Die ARexx-Variablen

```

end
say 'Waiting until ' || hours || ':' || minutes
address command 'wait until ' || hours || ':' || minutes /* warten */
'QUERY FROZEN'
if result='TRUE' then 'FREEZE' /* FrameGrabber aktivieren */
address command wait 1 secs
'CALCPICLEVEL'
'QUERY SAVEAS'
if result='SCREEN' then 'RENDER'
'SAVE'
'QUERY FILENAME'
say 'Picture saved as: ' || result
end

```

### 3.3 Was ist noch zu tun?

Dies ist die nicht vollständige Liste der noch nicht verwirklichten Möglichkeiten der Software:

- Fehler beseitigen
- Farbfilter verbessern (evtl. im Frequenzbereich?)
- HAM-Modus optimieren
- Synchronisation verbessern
- Graduations-Entzerrung (Gamma-Regler)
- Meßtechnik: Oszilo- und Vektorskop, Spectrum-Analyser
- Auslesen der Zusatzinformation in der Austastlücke (Videotext, VPS, Zeile 23)
- PALplus (inklusive 16:9-Screen)
- NTSC, SECAM, MAC, ...
- Descrambler
- Geschwindigkeitsoptimierungen
- ...

### 3.4 Fehler

Fehlerberichte erreichen mich am besten per EMail: [h.boehme@tu-bs.de](mailto:h.boehme@tu-bs.de).

Wenn ich weitere Videonormen (NTSC oder SECAM) in die Software integrieren soll, schickt mir gepackte `.video`-Files. Wenn ich Grafikkarten unterstützten soll, schickt mir eine Grafikkarte (oder zumindest deren Dokumentation).

#### 3.4.1 Bekannte Fehler

- Die SEL-Leitung ist Amigaseitig mit RI der seriellen Schnittstelle verbunden. Das kann zu Problemen mit Modemverbindungen führen → Kurzfristige Abhilfe: Den Framegrabber abklemmen, wenn man das Modem benutzen will, langfristig ist die Hardware zu modifizieren.

## Literatur

- [1] Fernsehtechnik und Bildübertragung (Teil I & II), Prof. Dr.-Ing. U. Reimers, Vorlesungsscripte, Institut für Nachrichtentechnik, Technische Universität Braunschweig
- [2] Digitale Signalverarbeitung in Fernsehgeräten, Paschko, Verlag Technik
- [3] Von PAL zu PALplus, Dobler, Verlag Technik
- [4] Digitale Filter in der Videotechnik, Schönfelder (Hrsg.), Drei R
- [5] Computer-Sehen, Prof. Dr. Friedrich M. Wahl, Vorlesungsscript, Institut für Robotik und Prozeßinformatik, Technische Universität Braunschweig
- [6] Digitale Bildsignalverarbeitung, Prof. Dr. Friedrich M. Wahl, Springer
- [7] Datenblatt TDA8708A, Phillips IC02,  
<http://www.semiconductors.philips.com/acrobat/2031.pdf>
- [8] Datenblatt HM530281R, Hitachi,  
[http://www.hitachi-eu.com/hel/ecg/pdf/01\\_063.pdf](http://www.hitachi-eu.com/hel/ecg/pdf/01_063.pdf)
- [9] Datenblatt LM1881, National  
<http://www.national.com/search/search.cgi/design?keywords=lm1881>
- [10] High-Speed CMOS Data, Motorola

## A Versionen

### A.1 Hardware

#### Revision 1.0

Erster Prototyp

#### Revision 1.1

**BUG** Videosignal des Filterausgangs darf nicht direkt mit SCART verbunden werden → Verstärker und Leitungsanpassung erforderlich!

**BUG** Böse Leitungseffekte, wenn die Schaltung mit einem Kabel vom nur 1,5m mit dem Amiga verbunden ist → Leitungsabschluß an Strobe (mit Tiefpaß). Außerdem dürfen die Clock-Eingänge nicht direkt mit Acknowledge verbunden werden (Rücklaufende Welle läßt nochmal Triggern!) → Zweiter Inverter erzeugt das Signal Separat!

**BUG** Der LM1881-Syncseparator hat große Probleme mit verrauschtem Videosignal, er arbeitet auch sonst etwas unregelmäßig. → Tiefpaß einfügen!

**BUG** Zu Dumm – Der TDA8708 verlangt als Eingabe für GateA und GateB Active High Signale, CSync und Burst vom LM1881 sind aber Active Low. Folge: Der TDA ist die ganze Zeit im Controlmode 1 und die digitalisierten Bilder sind in der Regel etwas übersteuert! → Inverter einfügen!

Außerdem muß GateB gegenüber GateA verzögert werden (um sagen wir mal eine Mikrosekunde), dann (und nur dann) geht's.

**NEU** PNP-Transistorschaltung mit BUSY verbunden → Softwaremäßiges Ein- und Ausschalten ist jetzt möglich.

## A.2 Software

### Version 1.0

Erste Veröffentlichung

### Version 1.1

**NEU** Lokalisation

**NEU** AppWindow und AppIcon

**NEU** Kommandozeile und Menü

**NEU** ARexx-Port

### Version 1.2

**VERBESSERT** Parallelport: Die Interruptroutine wurde (optional) ersetzt durch eine Burst-routine. → Preview ist jetzt doppelt so schnell.

**NEU** Rauschfilter

**NEU** Farbflankenversteilerung

**BUG** Glätten der Vorschau war fehlerhaft

**VERBESSERT** Der Farbträgerbandpaß ist jetzt besser abgestimmt und mehrfach kaskadiert  
→ Cross-Color Störungen verringert

**VERBESSERT** Kammfilter kommt jetzt mit vertikalen Farbkontrasten besser zurecht.

## B Die Schaltpläne

Abbildung 12: Framegrabber Schaltplan, Seite 1





Abbildung 13: Framegrabber Schaltplan, Seite 2

## **C Das Platinenlayout**

Abbildung 14: Framegrabber Oberseite (gespiegelt)

---

Abbildung 15: Framegrabber Unterseite (gespiegelt)

## **D Der Bestückungsplan**

Abbildung 16: Framegrabber Bestückungsplan

## E Die Bauteileliste

Bauteil	Wert/Name	Gehäuse	Anzahl
B1	B80Cxxxx	WM	1
C1	2200 $\mu$	ES-5	1
C2,C3	100 $\mu$	ES-2,5	2
CON1	CLAMP2	CLAMP-02	1
CON2	SCART90M	SCART	1
CON3-CON8	CINCH	CINCH	6
CON9	D-25M90°	D-25M90°	1
CS1,CS2,CS14-CS21,CS24	100n	1206	11
CS3-CS5	4 $\mu$ 7	1206	3
CS6	18n	1206	1
CS7,CS25	220n	1206	2
CS8,CS13,CS23	1 $\mu$	1206	3
CS9	10p	1206	1
CS10,CS11	15p	1206	2
CS12	10n	1206	1
CS22	1n	1206	1
CS26	510p	1206	1
D1	LED	LED	1
D2	P6KE6V8CP	D-10	1
D3,D4	LC-LED	LED	2
F1	50mA	MINIFUSE	1
IC1	LM1881	DIL8	1
IC2	TDA8708A	SO28	1
IC3,IC4	7805S	TO220A	2
IC5,IC6	HM530281R	TSOP44	2
IC7	TLP3042	DIL6.CUT	1
IC8,IC12	74HC04	SO-14	2
IC9	74HC175	SO-16	1
IC10	74HC74	SO-14	1
IC11	74HC32	SO-14	1
J1	CON03X2	CON03X2	1
LS1	82 $\mu$ H	1210	1
LS2	1 $\mu$ H	1210	1
OSC1	16MHz	OSC314	1
RS1	680k	1206	1
RS2-RS4,RS16	75	1206	4
RS5	330	1206	1
RS6	680	1206	1
RS7,RS13,RS14	2k2	1206	3
RS8,RS10	470	1206	2
RS9	100k	1206	1
RS11,RS12,RS18	1k	1206	3
RS15	150	1206	1
RS17	620	1206	1
RS19,RS20	10k	1206	2
TR1	2×3V, 666mA	FTR4-6	1
TS1	NPN(848)	SOT-23	1
TS2	PNP(858)	SOT-23	1
VAR1	300V	VAR-5	1
PCB1	–	Platine	1
PCK1	Strapu-Gehäuse	Typ 2063	1

Tabelle 3: Die Framegrabber Bauteileliste