

Scanned  
by  
105th

# MacroSystem

---

V L A B 4.0

Y U V -  
Echtzeit -  
Video -  
Digitizer

für den  
Commodore Amiga

---

**Benutzerhandbuch**

**MacroSystem**

---

V L A B 4.0

YUV-  
Echtzeit-  
Video-  
Digitizer

für den  
Commodore Amiga

---

**Benutzerhandbuch**

# VLAB — YUV-Echtzeit-Video-Digitizer für den Commodore-Amiga

Deutschsprachiges Anwenderhandbuch — sechste Auflage, Juni 1993 (Version 4.0)

This manual is the German version of the *VLab* documentation. If you prefer English, Netherlands or French, please contact your dealer or MacroSystem!

*VLab* Hardware: Hartmut Sprave

*VLab* Software: Henning Friedl und Edwin Bielawski

*VLab* Handbuch: Martin Sprave, Ulrich Hirner, Henning Friedl und Edwin Bielawski

*VLab* Platinenlayout: Bernd Gronemann

*VLab* Piktogramme: Frank Emmerling und Martin Sprave

## Urheberrecht

*VLab* © Copyright 1992/93 by MS MacroSystem Computer GmbH, D-5810 Witten

Alle Rechte, insbesondere das Recht auf Vervielfältigung, Verbreitung und Übersetzung vorbehalten. Kein Teil dieses Werkes darf ohne ausdrückliche schriftliche Genehmigung der Autoren vervielfältigt oder auf Datenträger gespeichert werden.

Beachten Sie, daß selbstverständlich auch Videobilder dem Urheberrecht unterliegen! Klären Sie daher vor einer kommerziellen Verwendung von Bildmaterial alle urheberrechtlichen Fragen!

## Nutzungsrecht

Das Programm darf nur auf dem Rechner eingesetzt werden, in dem die zugehörige Hardware installiert ist. (Ein Ausnahme besteht im Gebrauch von Demo-Versionen.) Somit ist das Nutzungsrecht entsprechend der tatsächlichen Möglichkeit der Benutzung eines Buches beschränkt.

Diskettenkopien dürfen lediglich zum Zwecke der Datensicherung angefertigt werden. Der Nachbau der Hardware und die Reproduktion des Handbuches (auch auszugsweise) sind nicht erlaubt.

## Haftung

Dieses Produkt wurde mit großer Sorgfalt hergestellt. Trotzdem sind Fehler nie ganz auszuschließen. Es kann daher keine Gewähr dafür übernommen werden, daß *VLab* unterbrechungs- oder fehlerfrei abläuft und daß die enthaltenen Funktionen in allen von Ihnen gewählten Kombinationen ausführbar sind. Für die Erreichung eines bestimmten Verwendungszwecks wird ebenfalls keine Gewähr übernommen. Die Haftung für unmittelbare Schäden, mittelbare Schäden, Folgeschäden und Drittschäden ist, soweit gesetzlich zulässig, ausgeschlossen. Die Haftung bei grober Fahrlässigkeit und Vorsatz bleibt hiervon unberührt, in jedem Fall ist jedoch die Haftung auf den Kaufpreis beschränkt.

Der Inhalt dieses Handbuches kann ohne Ankündigung geändert werden und ist nicht als eine Garantieerklärung anzusehen. Technische Änderungen bleiben vorbehalten.

## Warenzeichen

*VLab* und *Retina* sind Warenzeichen der MacroSystem US Inc.

*Amiga* ist ein eingetragenes Warenzeichen der Commodore-Amiga Inc.

*Amiga 500, 600, 1200, 2000, 3000, 4000, CDTV, AmigaDOS, Amiga Workbench, Amiga Kickstart* und *AutoConfig* sind Warenzeichen der Commodore-Amiga Inc.

*Commodore*, das *Commodore-Firmensymbol* und *CBM* sind eingetragene Warenzeichen der Commodore Electronics Ltd.

*Art Department Professional* und *Morph Plus* sind eingetragene Warenzeichen der ASDG Inc.

Diese Dokumentation kann auch weitere Warenzeichen enthalten, von denen angenommen wird, daß sie Eigentum der jeweiligen Firmen sind.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>8</b>
<b>2</b>	<b>Kurzeinführung in VLab</b>	<b>10</b>
<b>3</b>	<b>Installation</b>	<b>13</b>
3.1	Voraussetzungen . . . . .	13
3.2	Installation der Hardware . . . . .	13
3.3	VLab/par: die externe Version . . . . .	15
3.4	Numerierung der Eingangsbuchsen . . . . .	16
3.5	Installation der Software . . . . .	17
<b>4</b>	<b>Die Software</b>	<b>18</b>
4.1	Die Oberfläche von VLab . . . . .	18
4.1.1	Das Menü . . . . .	18
4.1.2	Die Fenster . . . . .	19
4.1.3	Die Gadgets . . . . .	19
4.1.4	Der File-Requester . . . . .	25
4.1.5	Der Font-Requester . . . . .	25
4.2	Die Titelleiste . . . . .	26
4.3	Der Mauszeiger . . . . .	26
4.4	Langzeitoperationen . . . . .	27
4.5	Die Speicherverwaltung . . . . .	27
<b>5</b>	<b>Die VLab-Funktionen im Detail</b>	<b>29</b>
5.1	Menü: Projekt . . . . .	29
5.1.1	Puffer löschen . . . . .	30

5.1.2	Laden	31
5.1.3	Speichern	31
5.1.4	Speichern als	31
5.1.5	Export	32
5.1.6	Datei löschen	33
5.1.7	Hilfe	34
5.1.8	Über VLab	34
5.1.9	VLab beenden	34
5.2	Menü: Fenster	35
5.2.1	Einlesen	35
5.2.2	Umrechnen	35
5.2.3	Sequenz	35
5.2.4	Film (IFR)	36
5.2.5	Farbkorrektur	36
5.2.6	Quelle wählen	39
5.2.7	Quelle definieren	40
5.2.8	Befehl ausführen	45
5.2.9	Fenster schließen	47
5.3	Menü: Steuern	48
5.3.1	Einlesen	48
5.3.2	Umrechnen	51
5.3.3	Sequenz	55
5.3.4	Film (IFR)	59
5.3.5	Bearbeiten	60
5.3.6	Monitor	62
5.3.7	Vorschau	64
5.3.8	Videoeingang	68
5.3.9	Hardware	69
5.4	Menü: Einstellungen	71
5.4.1	Icons speichern	71
5.4.2	ILBM komprimieren	71
5.4.3	Workbench öffnen	71

5.4.4	Retina-Screen-Modus fixieren . . . . .	72
5.4.5	Screen-Typ . . . . .	72
5.4.6	Zeichensätze . . . . .	74
5.4.7	Sprache . . . . .	76
5.4.8	Sicherheitsabfragen . . . . .	76
5.4.9	Einstellungen laden . . . . .	77
5.4.10	Einstellungen speichern . . . . .	78
5.5	Menü: Benutzer . . . . .	80
5.5.1	Makro laden . . . . .	80
5.5.2	Makro löschen . . . . .	81
5.5.3	Skript abbrechen . . . . .	81
<b>6</b>	<b>Praktisches Arbeiten</b> . . . . .	<b>82</b>
6.1	Der Monitor-Prozeß . . . . .	82
6.2	Die DeInterlace-Funktion . . . . .	83
6.3	Unterschiedliche Videoquellen . . . . .	84
6.4	Festlegung des Bildausschnittes . . . . .	85
6.5	Sequenzaufnahmen . . . . .	86
6.6	Interleaved Frame Recording . . . . .	87
6.6.1	Allgemeines . . . . .	87
6.6.2	Wie funktioniert IFR? . . . . .	88
6.6.3	Die einzelnen Bedienelemente . . . . .	89
6.6.4	Durchführung einer Aufnahme . . . . .	92
6.6.5	Erstellen einer Animation . . . . .	95
6.6.6	Tips zu IFR-Problemen . . . . .	95
6.7	Die Farbkorrektur . . . . .	96
6.7.1	Farbe . . . . .	97
6.7.2	Rot,Grün,Blau . . . . .	97
6.7.3	Kontrast . . . . .	98
6.7.4	Helligkeit . . . . .	98
6.7.5	Gamma . . . . .	98
6.8	Probleme mit Amiga IFF-ILBM-Dateien . . . . .	99

6.8.1	Kompression . . . . .	99
6.8.2	Bildbreite . . . . .	99
6.8.3	Farbpalette . . . . .	99
<b>7</b>	<b>Etwas Videotechnik</b>	<b>100</b>
7.1	Das FBAS-Signal . . . . .	100
7.2	Was bedeutet eigentlich „YUV“? . . . . .	101
7.3	Braucht man für VLab eine Grafikkarte? . . . . .	102
7.4	Wird ein „Time-Base-Corrector“ benötigt? . . . . .	103
7.5	Was wird unter „S-VHS“ verstanden? . . . . .	103
7.6	Die Hardware-Konzeption von VLab . . . . .	104
7.7	Das PAL-System . . . . .	105
7.7.1	Die Schwarzweiß(Y)-Auflösung des PAL-Systems . . . . .	105
7.7.2	Die Farbkodierung im PAL-System . . . . .	105
7.7.3	Die Farbauflösung des PAL-Systems . . . . .	107
<b>8</b>	<b>Makros und ARexx</b>	<b>108</b>
8.1	Allgemeines . . . . .	108
8.2	Beispiel für ein Vlab-Makro . . . . .	109
8.3	Der CLI-Befehl VLabCmd . . . . .	110
8.4	Mitgelieferte Makros . . . . .	111
8.4.1	Edit.rx . . . . .	111
8.4.2	Edit.sc . . . . .	111
8.4.3	ParseSequence.rx . . . . .	111
8.4.4	ScanTestLines.vm . . . . .	111
8.4.5	SequenceToAmiga.rx . . . . .	112
8.4.6	SetSpecialValues.rx . . . . .	112
8.4.7	VLABtoYUVN.rx . . . . .	113
8.4.8	VShell.rx . . . . .	113
8.4.9	LoMem.vm und HiMem.vm . . . . .	113
8.5	Kurzbeschreibung der Parameterarten . . . . .	113
8.6	Rückgabewerte und Fehlercodes . . . . .	114
8.7	Die einzelnen ARexx-Befehle . . . . .	115

<b>9</b>	<b>Verschiedenes</b>	<b>159</b>
9.1	Die ADPro-Module . . . . .	159
9.1.1	Der VLAB Image-Grabber . . . . .	159
9.1.2	Der YUVN Image-Loader . . . . .	160
9.2	Tips und Fehlerhilfen . . . . .	162
9.3	Über Public Screens . . . . .	166
9.4	Internes für Programmierer . . . . .	166
9.5	Die Tastatur-Kurzbefehle (Short Cuts) . . . . .	167



# Kapitel 1

## Einleitung

*VLab* (sprich Fau-Lab oder Wie-Läb) bietet dem Amiga-Anwender die Möglichkeit, hochqualitative Videobilder aus laufenden Videoquellen (Fernsehempfänger, Videorecorder, Bildplattenspieler, Kabeltuner, Videokamera...) einzulesen zu können.

*VLab* digitalisiert dabei ständig in Echtzeit (Real-Time) die Videodaten, die am ausgewählten Videoeingang anliegen. Auf Anforderung werden die digitalen Daten dann in den auf der Karte befindlichen Zwischenspeicher eingelesen. Diese Daten im *VLab*-RAM können dann sofort vom Amiga abgerufen werden.

Zur Steuerung des Gerätes wird eine sehr umfangreiche und äußerst komfortable Software mitgeliefert, die konzeptionell auf modernstem Stand ist. MacroSystem hat als echte Pioniertat die Lauffähigkeit der Software vom Vorhandensein der Kickstart 2.0 (V37.175 oder höher) abhängig gemacht. Die wirklich überragenden Fähigkeiten des neuen Amiga-Betriebssystems konnten so voll in der *VLab*-Software Verwendung finden. Weil die zusätzlichen Funktionen der neuen Systemsoftware in der Commodore-Dokumentation ausführlich beschrieben sind, wird ihre Kenntnis in diesem Handbuch vorausgesetzt.

Zur Beobachtung der Videodaten auf dem Amiga-Monitor kann das Videobild fortlaufend mit einer relativ hohen Bildrate in einem Amiga-Fenster dargestellt werden. Dieses Bild arbeitet mit 2-16 Graustufen und einer verkleinerten Darstellung. So kann der Zeitpunkt eines „Schnappschusses“ optimal abgepaßt werden. Per Mausclick (oder per ARexx-Kommando) wird dann der Einlesevorgang in maximaler Qualität durchgeführt. Die Bilddaten liegen nun im YUV-Format vor und können zunächst durch eine Vorschau-Funktion kontrolliert und anschließend in unterschiedlichste Formate umgerechnet und angezeigt werden.

Weiterhin ist es möglich, die Bilder fortlaufend einzulesen und auf einem frei wählbaren Medium (RAM-Disk, Festplatte etc.) in höchstmöglicher Geschwindigkeit aufeinanderfolgend abzuspeichern — ein Feature, welches von allen professionellen Fotografen („Winder-Kamera“) angewandt wird, um hinterher das beste „Foto“ einer Serie zu

ermitteln.

Dargestellt werden kann das Bild entweder in einem dem Amiga bekannten Format (HAM, HiRes etc.) oder direkt auf einer 24/32 Bit-Grafikkarte (MacroSystem *Retina* oder ACS Harlequin). Die Darstellung in den Amiga-Modi bedeutet dabei natürlich eine Qualitätsabsenkung, da die volle Farbzahl und Auflösung des YUV-Bildes nicht ohne eine Grafikkarte darstellbar ist. Dennoch ist die Qualität schon auf normalen Amigas wirklich gut und auf dem Amiga 4000 begeisternd.

*VLab* bietet vollen ARexx-Support und kann damit auch von anderen Anwendungen aus gesteuert werden. Zahllose ARexx-Befehle sind in der *VLab*-Software implementiert, um ein breites Spektrum externer Anwendungsmöglichkeiten zu ermöglichen.

Das vorliegende Handbuch wurde wie folgt gegliedert: Nach einer Kurzeinführung für den ungeduldigen Leser wird zunächst in Kapitel 3 die Installation der Hard- und Software beschrieben. Danach werden in Abschnitt 4 allgemeine Informationen über den Aufbau des Programms gegeben. Ab Seite 29 findet man eine Auflistung sämtlicher *VLab*-Funktionen. Anschließend werden in Kapitel 6 verschiedene Aspekte der praktischen Arbeit mit *VLab* erläutert. Es folgt auf Seite 100 eine interessante Einführung in die Videotechnik. Der Abschnitt 8 befaßt sich mit *VLab*-Makros und ARexx. Schließlich kann man im Kapitel 9 unter „Verschiedenes“ z.B. wichtige Tips und Fehlerhilfen nachlesen.

Schon bei der Markteinführung im Frühjahr 1992 wurde die Direktunterstützung von *VLab* durch professionelle Amiga-Produkte zur Bildbearbeitung angekündigt. Mittlerweile existieren entsprechende Module für ADPro2 von ASDG (siehe Seite 159) und TVPaint von TecSoft. Auch wurde die Steuerung der Harlequin-Videokarte durch die *VLab*-Software verfeinert. Die Fähigkeiten der neuen MacroSystem-Grafikkarte *Retina* finden selbstverständlich auch schon Verwendung. Das für *VLab* eingeführte Dateiformat „YUVN“ ist bei Commodore als IFF-Standard registriert worden und wird inzwischen sogar von Konkurrenzprodukten verwendet.

Das *VLab*-System, insbesondere die Software, unterliegt ständigen Veränderungen (zahlreiche Verbesserungen wurden bereits verwirklicht). Für die Entwickler ist es dabei wichtig zu erfahren, welche Funktionen der Anwender im praktischen Einsatz noch vermißt. Für Fehlermeldungen und Verbesserungsvorschläge wären wir deshalb im Interesse aller Kunden besonders dankbar. Beachten Sie jedoch, daß *VLab* ein Projekt zur Digitalisierung von Videobildern bleiben soll — die Neugestaltung zu eigenen Grafiken möchten wir speziellen Bildbearbeitungsprogrammen überlassen.

Die jeweiligen Software-Updates sind gegen einen geringen Betrag (per Scheck Vorkasse) und Einsendung der alten Diskette bei MacroSystem direkt erhältlich.

Ihr MacroSystem-Team wünscht Ihnen viel Erfolg bei der Arbeit mit *VLab*!

# Kapitel 2

## Kurzeinführung in VLab

Bauen Sie als erstes die *VLab*-Hardware in Ihren Rechner ein. Eine ausführliche Beschreibung des Einbauvorgangs können Sie Kapitel 3.2 entnehmen.

Verbinden Sie den Ausgang Ihrer Signalquelle (mit Hilfe eines passenden Videokabels) mit einer der *VLab*-Eingangsbuchsen. Die softwareseitige Zuordnung der Anschlüsse wird dazu in Abschnitt 3.4 beschrieben.

Nachdem Sie das *VLab*-Programm auf Ihrer Festplatte installiert haben, können Sie beginnen, mit *VLab* zu arbeiten. Näheres zur Installation der Software finden Sie in Kapitel 3.5. Sie können aber auch direkt von der *VLab*-Diskette booten und sofort mit der Arbeit beginnen.

Starten Sie *VLab* durch einen Doppelklick auf das *VLab*-Icon in der *VLab*-Schublade. Je nach abgespeicherten Voreinstellungen öffnet *VLab* nun seinen eigenen Screen oder startet auf der Workbench. Für diese kurze Einführung wollen wir einen eigenen Screen verwenden. Falls *VLab* auf der Workbench gestartet ist, drücken Sie die rechte Maustaste, begeben sich in das Menü *Einstellungen* und wählen unter *Screen-Typ* die gewünschte Bildschirmauflösung. Besitzt Ihr Rechner einen *FlickerFixer*, empfiehlt sich die Einstellung *HiRes-Interlaced*, ansonsten die Betriebsart *HiRes*, hier ist jedoch die Wiedergabequalität beim Monitorbetrieb, aufgrund der geringeren vertikalen Auflösung, schlechter als in den anderen Modi. Besitzer von *ECS (Enhanced Chip Set)*, aber ohne *FlickerFixer*, können auch den *Productivity*-Modus wählen. In diesem Modus können maximal vier Farben gleichzeitig dargestellt werden. Stellen Sie anschließend in dem gleichen Untermenü die Anzahl der Farben auf 8, bzw. 4 für *Productivity*.

Beim Arbeiten im Interlace- oder Productivity-Modus ist die Standard-Schriftart der Workbench (Topaz 8) etwas schwer zu lesen. Wählen Sie in diesem Fall unter Einstellungen » Zeichensatz die Schrift Topaz 11, welche für den Einsatz auf Interlace-Screens gedacht ist.

Als nächstes werden über das Menü Fenster die Fenster Einlesen und Umrechnen geöffnet, sofern diese noch nicht offen sind.

Jetzt sind wir bereit, um mit dem Einlesen eines Bildes zu beginnen. Damit wir sehen können, was wir digitalisieren, starten wir über Steuern » Monitor » Betrieb die Monitorfunktion.

Wenn Sie Ihre Videoquelle an Buchse 1 angeschlossen haben, müßten Sie sofort ein Bild erhalten. Haben Sie die Videoquelle an Buchse 2 angeschlossen, müssen Sie erst über Steuern » Videoeingang auf Eingang 2 umstellen. Falls Sie noch immer kein Bild erhalten, überprüfen Sie bitte Ihre Verkabelung und ob Ihre Videoquellen eingeschaltet sind und ein Signal liefern.

Wird im Monitor-Fenster ein Bild angezeigt, können Sie es durch Druck auf den Start-Button im Einlesen-Fenster digitalisieren und in den Speicher des Amiga übertragen. Um einen schnellen Eindruck von dem digitalisierten Bild zu erhalten, können Sie die Vorschau-Funktion einsetzen. Wählen Sie dazu die gewünschte Darstellungsart über Steuern » Vorschau. Für den Anfang sollten Sie auf Skaliert und Fenster schalten. Die Vorschau erfolgt nun in einem Fenster auf dem VLab-Screen mit der entsprechenden Anzahl an Graustufen. Die Option Farbe ist hier wirkungslos, auf dem VLab-Screen wird die Vorschau immer in Schwarzweiß angezeigt. Die Größe des Vorschau-Fensters können Sie über das Sizing-Gadget rechts unten selbst bestimmen.

Nun erhalten Sie schon einen groben Eindruck von dem zuvor eingelesenen Bild. Entspricht dieses noch nicht Ihren Vorstellungen, können Sie durch Druck auf den Start-Button sofort ein neues Bild einlesen und durch Drücken von Vorschau anzeigen lassen. Die Anzeige des Vorschau-Bildes nach dem Einlesen eines Bildes geschieht automatisch, wenn in Steuern » Vorschau die Option Vorschau automatisch gesetzt wurde.

Sollte in dem Vorschau-Fenster nur ein Teil des Monitorbildes sichtbar sein und die Option Steuern » Vorschau » Skaliert ist gesetzt, ist es möglich daß eine spezielle Quellendefinition gewählt ist. Öffnen Sie dann mit Fenster » Quelle wählen das Fenster für die Auswahl einer Quellendefinition. Wählen Sie dort die Definition Standard und drücken Sie den OK-Button. Nun müßte nach nochmaliger Digitalisierung und Starten der Vorschau praktisch das gesamte Monitorbild im Vorschau-Fenster sichtbar sein.

Unter Umständen sind im Vorschau-Bild eigenartige Verzerrungen erkennbar, die sich als Flackern oder als kammförmige Versetzungen zeigen. Diese resultieren aus der Tatsache, daß Vollbilder beim Fernsehen als zwei ineinander verschachtelte Halbbilder gesendet werden, die bei der Aufnahme auch zu zwei kurz aufeinanderfolgenden Zeitpunkten aufgezeichnet wurden. Objekte in Bewegung können sich jedoch in dieser Zeitspanne schon merklich weiterbewegt haben, so daß dieser Effekt entsteht. Mit **Steuern** » **Bearbeiten** » **DeInterlace** lassen sich solche Phänomene zuverlässig beseitigen.

Das eingelesene Bild kann nun mit **Projekt** » **Speichern** als YUV-Bild oder mit **Projekt** » **Export** » **24 Bit ILBM** als 24 Bit RGB-Bild abgespeichert werden.

Um ein hochwertiges Amiga-Bild aus unserem digitalisierten Bild zu erhalten, müssen Sie sich dem **Umrechnen**-Fenster zuwenden. In dem **Listview-Gadget** links werden die zur Verfügung stehenden Farbmodi angezeigt. Wählen Sie einen davon aus und drücken Sie den **Start-Button**. Der Rechner ist kurz beschäftigt, was durch einen veränderten Mauszeiger verdeutlicht wird. Meist erscheint auch ein Fenster, welches auf eine Langzeitoperation aufmerksam macht. Daraufhin kann die Anzeige des berechneten Bildes durch Druck auf den **Anzeige-Button** ausgelöst werden. Das Amiga-Bild wird nun auf einem separaten Screen angezeigt. Dieser kann über sein Menü oder je nach eingestellter Option über den rechten Mausknopf wieder geschlossen werden.

Für den Anfang erhält man mit dem **HAM-Modus** mit 4096 Farben in **LoRes** die besten Ergebnisse für farbige Bilder. Detailreicher wird ein Schwarzweißbild mit 16 Graustufen in **HiRes**. In hoher Horizontalauflösung kann der Amiga jedoch maximal 16 Farben oder Graustufen darstellen. Mit der Option **Floyd-Steinberg** kann eine Schattierung (**Dithering**) zugeschaltet werden. Diese steht jedoch nicht im **HAM-Modus** zur Verfügung. Bilder die in den Sondermodi berechnet werden, können auf dem Amiga zwar nicht dargestellt, wohl aber abgespeichert werden.

Das umgerechnete Bild können Sie mit der Funktion **Projekt** » **Export** » **Amiga** als **Amiga-IFF-Bild** abspeichern.

Die Kurzeinführung in das **VLab-Programm** ist nun beendet. Spielen Sie ruhig ein wenig mit den verschiedenen Optionen und Einstellungen, um alle Möglichkeiten auszuschöpfen, die das Programm bietet. Ausführliche Erläuterungen zu allen Funktionen erhalten Sie im Kapitel 5 „Die **VLab-Funktionen** im Detail“.

# Kapitel 3

## Installation

### 3.1 Voraussetzungen

Zum Betrieb des *VLab*-Echtzeit-Video-Digitizers benötigt man:

- Einen Commodore-Amiga
- Videoquellen mit FBAS- bzw. Y/C-Ausgang:  
Fernsehgeräte, Videorecorder, Videokameras, Kabeltuner...
- Geeignete Verbindungskabel
- Die Amiga-OS2-Systemsoftware (ab Kickstart V37.175, Workbench V37.67)  
Benutzen Sie bitte nur offiziell erschienene Versionen! Neuere Workbench-  
Releases werden auch unterstützt.

Als Mindestsystemausbau werden 1 MB Chip-Memory, 2 MB Fast-Memory und eine 50 MB SCSI-Festplatte empfohlen. Sinnvolle Erweiterungen sind etwa ein Display-Enhancer wie die MacroSystem DeInterlace-Karte (für A2000) sowie eine zusätzliche Grafikkarte (z.B. *Retina* von MacroSystem oder „*Harlequin*“ von ACS).

### 3.2 Installation der Hardware

**Achtung:** Schalten Sie als erstes Ihren Amiga und alle Zusatzgeräte aus und ziehen Sie dann den Netzstecker vom Amiga ab!

Der Ein- oder Ausbau von Hardware bei eingeschaltetem Gerät kann sowohl Ihren Amiga, als auch die Erweiterung beschädigen!

Führen Sie die nachfolgenden Schritte genau durch und beachten Sie auch die Hinweise in den Handbüchern Ihrer Videogeräte!

- Ziehen Sie das Netzkabel sowie den Stecker der Maus und der Tastatur vom Rechner ab!
- Lösen Sie die Schrauben, mit denen das Gehäuseoberteil befestigt ist! Je zwei davon befinden sich unten an beiden Seiten und eventuell eine weitere oben in der Mitte der Rückwand.
- Fassen Sie den Deckel an den Seiten und ziehen Sie ihn behutsam ab. (Beim A 2000 nach vorn, beim A 3000/4000 nach hinten!) Heben Sie ihn dann vorsichtig nach oben weg!
- Wählen Sie einen beliebigen, freien, 100-poligen Erweiterungssteckplatz aus! Von diesen sind beim Amiga 2000 im vorderen Teil der Platine fünf Stück vorhanden. Beim Amiga 3000/4000 befinden sich vier Expansionslots auf einem sogenannten „Daughterboard“, das senkrecht zur Hauptplatine steht.
- Entfernen Sie das zum Steckplatz gehörige Blindblech auf der Rückseite des Gehäuses!
- Setzen Sie beim A 2000 die VLab-Karte mit den Bauteilen Richtung Laufwerksträger (also nach rechts zeigend) ein. Beim A 3000/4000 muß die Bestückungsseite nach oben weisen. Wenden Sie dabei leichten Druck an und achten Sie darauf, daß die Platine bis zum Anschlag im Slot sitzt!
- Schrauben Sie das Abschlußblech der VLab-Platine mit der vorhandenen Schraube an der Gehäuserückwand fest!
- Jetzt können Sie das Rechnergehäuse wieder schließen.

Da die VLab-Hardware über das Amiga-typische AutoConfig verfügt, brauchen keinerlei Adreßeinstellungen vorgenommen zu werden.

Nun können Sie den Digitizer mit den Ausgängen Ihrer Videoquellen verbinden. Auf der VLab-Seite benötigt man dazu einen Cinchstecker. Ihre Videogeräte werden eventuell einen anderen Anschluß, wahrscheinlich den Euro-AV-Stecker (SCART), besitzen. Dafür geeignete Adapterkabel sind in großer Auswahl im Fernseh-Fachhandel erhältlich. Verwenden Sie jedoch bitte keine Audiokabel, weil diese nicht die für Videosignale erforderliche Schirmung aufweisen! Die Anschlußbelegung für die verschiedenen VLab-Versionen findet man auf Seite 16.

**Vorsicht: Der Cinchanschluß kann leicht mit einem Antennenstecker verwechselt werden. Verbinden Sie niemals ein Antennenkabel direkt mit Ihrer VLab-Hardware, sondern nur über einen Fernseh-Tuner! Eine Zerstörung des Video-Digitizers wäre sonst die Folge.**

### 3.3 VLab/par: die externe Version

Die Version *VLab/par* in Form eines externen Gerätes wird über die parallele Drucker-schnittstelle (Centronics) angeschlossen und ist daher für alle Amiga-Modelle gleichermaßen geeignet. Insbesondere bei den kleinen Modellen ohne Zorro II-Steckplätze (A500, A500+, A600) sowie bei Systemen, bei denen alle Erweiterungs-slots schon belegt sind, wird diese Bauform benötigt.

**Vorsicht: Sämtliche Kabel dürfen nur montiert oder demontiert werden, wenn alle Geräte (Digitizer, Amiga, Drucker...) ausgeschaltet sind. Prüfen Sie auch, ob das mitgelieferte Netzgerät wirklich für die Spannung und die Frequenz Ihres örtlichen Stromnetzes geeignet ist!**

Der Anschluß der externen *VLab*-Version an Ihren Amiga ist denkbar einfach: Nach dem Ausschalten sämtlicher Geräte wird *VLab/par* mit dem beigelegten Verbindungskabel an die Druckerschnittstelle angeschlossen. Eventuell müssen Sie eine Umschaltbox verwenden, um einen Drucker weiterhin nutzen zu können.

**Achten Sie jedoch darauf, daß die Leitungslängen zwischen Rechner und Videodigitizer nicht zu groß werden! Auf keinen Fall sollten Sie weitere Geräte (Drucker) mit dem *VLab*-Kabel kurzschließen — verwenden Sie stets eine Umschaltbox, die alle Signale umschaltet!**

Nun kann man eine Videoquelle über ein Kabel mit einem der beiden Cincheingänge verbinden (wie in Abschnitt 3.2 beschrieben). Danach sollten Sie das beigelegte Stecker-netzteil zuerst an *VLab/par* und dann an das Stromnetz anschließen. Daraufhin kann man den Digitizer und den Rechner einschalten und die Software installieren.

Man sollte beachten, daß *VLab* den oben beschriebenen Mindestsystemausbau benötigt, um seine Leistungsfähigkeit voll entfalten zu können. Leider sind aus hardwaretechnischen Gründen die erzielten Datenraten über die Parallelschnittstelle wesentlich geringer als über den Zorro II-Bus. Die Übertragung digitalisierter Videobilder aus dem *VLab*-Video-RAM in den Speicher des Amiga dauert deshalb merklich länger als beim internen Modell. Trotzdem wird eine beachtliche Arbeitsgeschwindigkeit erzielt.

Bitte denken Sie daran, *VLab/par* jeweils vor dem Aufruf der Einlesesoftware einzuschalten. Die externe Hardware wird dann vom Programm automatisch erkannt und eingebunden. Sollten Sie dies versäumt haben, können Sie das Gerät nachträglich mit dem Menüpunkt *Steuern* » *Hardware* » *VLab/par* einbinden betriebsbereit melden.



### 3.4 Numerierung der Eingangsbuchsen

Mit Einführung der SVHS-Version mußte die Belegung der Videoeingänge verändert werden, weil nunmehr bis zu drei Signalquellen pro Hardware zur Auswahl bereitstehen. Um den Benutzer nicht zu verwirren, werden jetzt *jedem* Gerät softwaremäßig *immer* drei Eingangsnummern zugeordnet. Deshalb können ab VLab 3.0 nur noch zwei Geräte pro Rechner über das Menü angesteuert werden. Dabei belegt das erste VLab die Nummern 1...3 und der zweite Digitizer die Eingänge 4...6.

Der neu hinzugekommene Y/C-Anschluß ist in Form einer Mini DIN-Steckverbindung ausgeführt und befindet sich auf der jeweils höchsten Eingangsnummer (drei oder sechs). Es ist zu beachten, daß VLab/par immer zuletzt eingebunden wird und daher bei zwei Geräten die letzten drei Eingänge zugewiesen bekommt.

Die nachfolgenden Steckerbelegungen gelten bei Blickrichtung auf die Anschlußbuchsen.

Zuordnung der Signalquellen zu den drei Eingangsbuchsen				
VLab-Gerätetyp	Eingang Nr.	1 & 4 (FBAS)	2 & 5 (FBAS)	3 & 6 (Y/C)
Zorro II-Version im Amiga 2000		unten	oben	—
Zorro II-Version im A 3000/4000		links	rechts	—
VLab/par im externen Gehäuse		links	rechts	—
SVHS-Version im Amiga 2000		oben	mittig	unten
SVHS-Version im A3000/4000		rechts	mittig	links

**Hinweis:** Um bei hochwertigen Signalquellen (z.B. Hi8-Kamera) über die Y/C-Buchse die größtmögliche Bildqualität zu erhalten, müssen evtl. sämtliche Filter im Fenster „Quelle definieren“ abgeschaltet werden. Probieren Sie im Zweifelsfall verschiedene Einstellungen aus!

## 3.5 Installation der Software

Zunächst sollte man eine Sicherheitskopie der Original-Diskette anfertigen und anschließend auch nur diese benutzen.

Der nun folgende Transfer der *VLab*-Software auf die Festplatte Ihres Amigas<sup>1</sup> erfolgt recht komfortabel: Sie müssen lediglich einen Doppelklick auf dem Installationspiktogramm ausführen und dann das gewünschte Zielverzeichnis eintippen. Daraufhin werden automatisch die von *VLab* benötigten Verzeichnisse angelegt und die Software von der Diskette auf die Harddisk kopiert.

Seit der Softwareversion 3.1 wird das Commodore-Installer-Programm verwendet. Mit diesem Tool kann die Festplatteninstallation auf der grafikorientierten Benutzeroberfläche durchgeführt werden.

Nun können Sie das Programm aufrufen und sich mit dem Aufbau der *VLab*-Software vertraut machen. Öffnen Sie einige Programmfenster und „spielen“ Sie mit Funktionen wie Monitor, Einlesen, Vorschau, Umrechnen und Anzeige.

Nach wenigen Augenblicken sollten Sie schon die ersten Videobilder auf dem Bildschirm bewundern können. Gelingt Ihnen das nicht, haben Sie sicherlich den falschen Eingang angewählt, oder der Fehler steckt in der Verkabelung Ihrer Videogeräte mit der *VLab*-Hardware.

Hat man nach einer Weile seine persönliche Lieblingskonfiguration der *VLab*-Software herausgefunden, kann man sämtliche Einstellungen abspeichern und bei weiteren Programmstarts automatisch wiederherstellen lassen.

---

<sup>1</sup>Natürlich kann man das *VLab*-Programm auch immer direkt von Diskette starten.

# Kapitel 4

## Die Software

### 4.1 Die Oberfläche von VLab

*VLab* ist eines der ersten kommerziellen Programme, welches die Möglichkeiten der neueren Betriebssystemversionen OS 2.X voll ausnutzt. Viele Anwender werden sich daher mit der neuen Programmoberfläche, bestehend aus Menü, Fenstern und Gadgets, noch nicht sehr gut auskennen. In diesem Kapitel werden daher alle diese Elemente kurz erklärt. In den darauf folgenden Kapiteln wird dann davon ausgegangen, daß der Leser weiß, was z.B. mit einem Cycle-Gadget gemeint ist. Grundsätzliche Schwierigkeiten bei der Verwendung der neuen Amiga-Oberfläche sollten mit Hilfe des „Handbuchs zur Systemsoftware“ ausgeräumt werden.

#### 4.1.1 Das Menü

Viele Funktionen von *VLab* werden über das Pull-Down-Menü gewählt. Einige Funktionen lassen sich außerdem über Short-Cuts ohne Benutzung der Maus direkt von der Tastatur aus steuern. Short-Cuts, ein auch unter Kickstart 1.3 bekannter Begriff, sind die Tastaturkürzel zum Wählen einzelner Menüpunkte über eine Tastenkombination. Dabei hält man die rechte Amiga-Taste gedrückt, während man die Taste betätigt, die in dem Menü bei dem gewünschten Menüpunkt ganz rechts angegeben ist.

Bei der Menüwahl über die Maus ist zu beachten, daß *VLab* eine Mehrfachwahl erlaubt. Das bedeutet, daß der Bediener bei gedrückter rechter Maustaste das Menü durchgehen und alle gewünschten Punkte mit der linken Maustaste nacheinander anklicken kann. Erst nach Loslassen der rechten Maustaste werden dann alle gewählten Menüpunkte der Reihe nach abgearbeitet. Das ist besonders praktisch beim Verändern der Schalter, bei denen ein Häkchen vor dem Menüeintrag steht, sofern sie aktiviert sind.

Drei Punkte (...) hinter dem Menüeintrag symbolisieren, daß sich nach dem Aufruf ein Fenster öffnet — ein doppeltes Größer-Zeichen (>>) kennzeichnet das Vorhandensein von Untermenüs.

Wird während der Anwahl von Menüpunkten (bei gedrückter rechter Maustaste) die Help-Taste betätigt, erscheint ein Fenster mit einem Hilfetext über die jeweilige Funktion. Dazu muß sich jedoch eine spezielle Datei mit Texten in der aktiven Landessprache im Unterzeichnis *Catalogs* von *VLab* befinden.

### 4.1.2 Die Fenster

Das Programm *VLab* arbeitet in der Regel auf einem eigenen Screen (Bildschirm), auf dem verschiedene kleinere Fenster geöffnet werden können. Das Verschieben eines Fensters oder sein Schließen über das Close-Gadget (Schließsymbol) links oben in der Titelleiste ist wie unter Kickstart 1.3 möglich. Die beiden Gadgets in der oberen rechten Ecke der Titelleiste haben unter OS2.X allerdings eine etwas andere Bedeutung erhalten. Das Linke davon, sein Name ist Zoom-Gadget, verändert die Größe und Position des Fensters auf einen zweiten Wert. Bei *VLab* ist die zweite Größe fest eingestellt und entspricht genau den Abmessungen der Titelleiste. Größe, Position und Zustand aller *VLab*-Fenster werden beim Sichern der Einstellungen gespeichert und beim nächsten Programmstart übernommen. Das Gadget rechts neben dem Zoom-Gadget, sein Name lautet Depth-Gadget (Vorder-/Hintergrundsymbol), übernimmt die Funktion, das Fenster vor oder hinter andere Fenster zu bringen. Dabei wird bei Betätigung das Fenster nach vorne geholt, es sei denn, es war schon das vorderste, dann wird es nämlich ganz nach hinten gebracht. Das letzte Gadget ist das Sizing-Gadget in der Ecke rechts unten. Mit ihm läßt sich die Größe eines Fensters verändern.

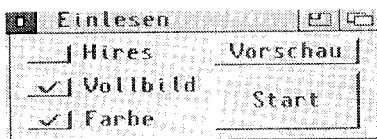


Abbildung 4.1: Das **Close**-Gadget

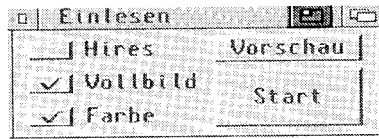


Abbildung 4.2: Das **Zoom**-Gadget

### 4.1.3 Die Gadgets

Unter OS2.X werden dem Programmierer (und damit natürlich auch dem Anwender) vom Betriebssystem (genauer der Gadtools-Library) eine Vielzahl verschiedener Sorten von Gadgets zur Verfügung gestellt. Jeder dieser Gadget-Typen hat ein festgelegtes Aussehen und einen bestimmten Verwendungszweck. Dadurch ist es möglich, daß die

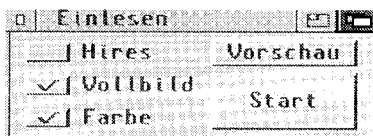


Abbildung 4.3: Das **Depth-Gadget**



Abbildung 4.4: Das **Sizing-Gadget**

Oberflächen vieler Amiga-Programme einheitlicher aussehen und deshalb vom Benutzer wesentlich leichter und schneller bedient werden können.

Die bei *VLab* benutzten Gadgets werden im folgenden kurz einzeln erklärt, damit der Anwender sie kennt und beim weiteren Studium dieser Anleitung die benutzten Namen und Begriffe versteht.

#### 4.1.3.1 **Button-Gadget**

Das Button-Gadget ist auch unter Kickstart 1.3 häufig eingesetzt worden und dürfte daher hinreichend bekannt sein. Sobald es angeklickt worden ist, wird nach Loslassen der linken Maustaste die zugehörige Aktion ausgelöst. Bewegt man bei gedrückter Maustaste den Mauszeiger vom Gadget weg, kann man die Maustaste loslassen, ohne daß die Aktion gestartet wird.

Zwei der Gadgets im *Einlesen*-Fenster sind Beispiele für Button-Gadgets. Bei der Betätigung eines der Gadgets wird eine Aktion gestartet, nämlich das Digitalisieren und Einlesen eines Bildes oder das Anzeigen einer Vorschau.

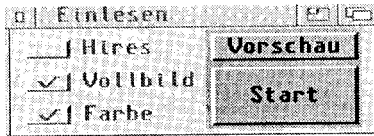


Abbildung 4.5: Zwei **Button**-Gadgets

#### 4.1.3.2 Checkbox-Gadget

Das Checkbox-Gadget stellt einen Schalter dar, der zwei Zustände annehmen kann: „Ein“ und „Aus“. Mit ihm werden in der Regel keine Aktionen gestartet, sondern verschiedene Voreinstellungen verändert, die eine später stattfindende Operation beeinflussen. Ist das Checkbox-Gadget „gesetzt“ oder „aktiv“, d.h. das Häkchen ist sichtbar, ist die Einstellung gewählt, die durch den Namen des Gadgets bestimmt ist. Ist das Häkchen nicht sichtbar, ist die Einstellung nicht aktiv, oder ihr Gegenteil kommt zur Anwendung.

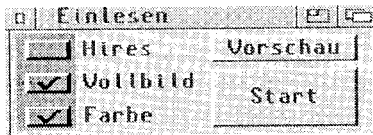


Abbildung 4.6: Drei **Checkbox**-Gadgets

Die Gadgets auf der linken Seite des Einlesen-Fensters sind Checkbox-Gadgets. Sie bestimmen die Betriebsart, die beim nächsten Digitalisieren Verwendung findet. Ist eine Option aktiv (Häkchen sichtbar), wird sie bei späteren Einlesevorgängen benutzt.

#### 4.1.3.3 Cycle-Gadget

Mit einem Cycle-Gadget kann aus einer Liste von mehreren Möglichkeiten eine ausgewählt werden. Wie auch beim Checkbox-Gadget wird damit eine Voreinstellung zur späteren Auswertung beeinflusst. Das Gadget sieht so ähnlich wie ein Button-Gadget aus, zeigt aber links einen gedrehten Pfeil. Bei jeder Betätigung des Gadgets wechselt der in ihm geschriebene Text, an dem man die aktuelle Stellung erkennen kann.

Im Fenster `Quelle definieren` befinden sich rechts drei solche Gadgets.

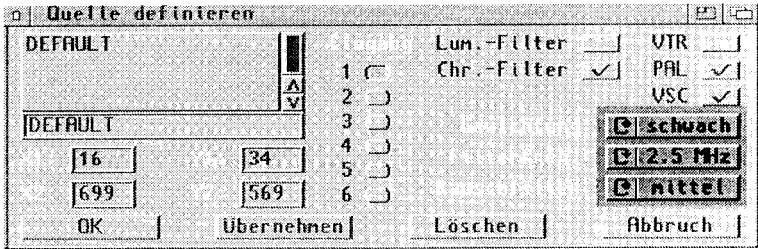


Abbildung 4.7: Drei Cycle-Gadgets

#### 4.1.3.4 Listview-Gadget

Listview-Gadgets (Roll-Listen) werden benutzt, um aus einer Auswahl von Möglichkeiten direkt die gewünschte anzuwählen. Ein Teil der Liste wird direkt angezeigt. Ein sichtbares Element kann durch Anklicken mit der Maus ausgesucht werden und wird dann unter dem angezeigten Teil der Liste eingetragen. Rechts befindet sich ein *Scroller-Gadget* (Rollbox). Es dient dazu, bei langen Listen den Ausschnitt zu wählen, der im Fenster als Auswahl sichtbar ist. Dazu ist entweder eins der kleinen Gadgets mit dem Rollpfeil mehrfach oder über einen längeren Zeitraum zu betätigen oder direkt der *Mover* (Rollbalken) zu verschieben.

Ein Listview-Gadget befindet sich z.B. neben zwei Button-Gadgets im Fenster *Quelle auswählen*.

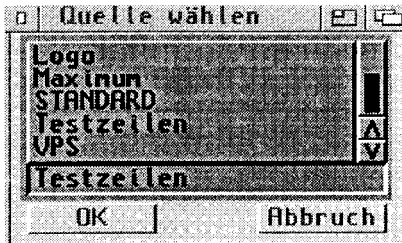


Abbildung 4.8: Das Listview-Gadget

#### 4.1.3.5 MX-Gadget

Eine weitere Möglichkeit, eine von mehreren Möglichkeiten auszuwählen, ist das MX-Gadget (Mutual eXclude = gegenseitiger Ausschluß). Das Gadget besteht aus mehreren

Elementen, von denen immer genau eins aktiv ist und damit den gewählten Zustand anzeigt. Im Fenster Quelle definieren wird ein MX-Gadget zur Auswahl eines Videoeingangs benutzt.

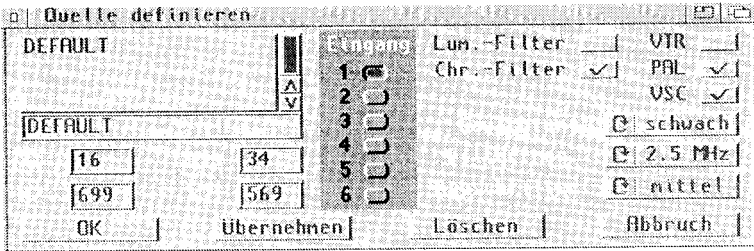


Abbildung 4.9: Das Mutual-Exclude-Gadget

#### 4.1.3.6 Slider-Gadget

Ein Slider-Gadget (Schieberegler) dient üblicherweise zum Einstellen eines Zahlenwertes. Wird der Mover bewegt, verändert sich der zugehörige Zahlenwert. Ein Mausklick neben den Mover verändert den Zahlenwert um einen Schritt. Das ist wichtig für exakte Einstellungen, wenn ein sehr hoher Wertebereich abgedeckt werden muß und eine schrittweise Einstellung nicht mehr anders möglich ist.

Im Farbkorrektur-Fenster werden über Slider-Gadgets die einzelnen Werte für die Farbkorrektur eingestellt.

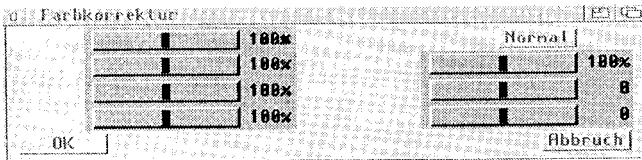


Abbildung 4.10: Sieben Slider-Gadgets

**Hinweis:** Bei der Ausgabe des Zahlenwertes kann es bei der Benutzung eines proportionalen Zeichensatzes zu leichten Ausgabefehlern kommen. Das ist kein Fehler von VLab, sondern ein Problem der Gadtools-Library.



### 4.1.3.7 String-Gadget, Integer-Gadget

String-Gadgets dienen zur direkten Eingabe von Zeichenfolgen über die Tastatur. Einmal mit der Maus aktiviert, kann ein beliebiger Text eingegeben werden. Paßt der Text nicht mehr in den sichtbaren Bereich, wird der Bereich seitlich verschoben. Wenn die maximale Länge des Textes überschritten wurde, blinkt der Bildschirm bei jedem Tastendruck einmal auf, um den Anwender zu informieren, daß keine Eingabe mehr erfolgen kann.

Integer-Gadgets sehen genauso aus und verhalten sich auch ebenso, mit der Ausnahme, daß man bei ihnen nur dezimale Zahlen eingeben kann. Werden unzulässige Tasten, z.B. Buchstaben, gedrückt, blinkt der Bildschirm einmal auf und die Eingabe wird ignoriert. Die Eingabe in einem String- oder Integer-Gadget kann, muß aber nicht, durch Drücken der Return- oder Enter-Taste abgeschlossen werden. Auch ein Druck auf die Tabulator-Taste beendet die Eingabe, mit dem Zusatz, daß bei Vorhandensein mehrerer String- oder Integer-Gadgets in diesem Fenster automatisch das nächste Gadgets zur Eingabe aktiviert wird.

Im Fenster `Befehl ausführen` befindet sich ein String-Gadget. Die vier kleinen Gadgets links unten im Fenster `Quelle definieren` sind Integer-Gadgets.

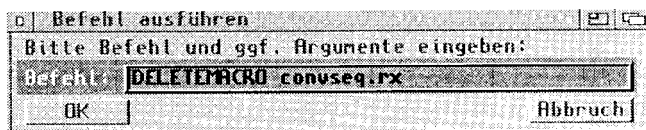


Abbildung 4.11: Ein String-Gadget

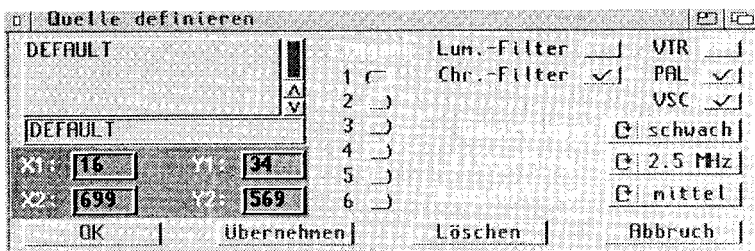


Abbildung 4.12: Vier Integer-Gadgets

#### 4.1.3.8 Text-Gadget, Number-Gadget

Text- und Number-Gadgets erlauben lediglich die Ausgabe von Texten oder Zahlen. Eine Eingabe ist hingegen nicht vorgesehen. Diese Gadgets erscheinen nur als Zeichenkette, die zusätzlich mit einem Rahmen umgeben sein kann. Sie ähneln dann den String- und Integer-Gadgets, mit dem Unterschied, daß im Gegensatz zu diesen hier keine Eingabe möglich ist.

#### 4.1.4 Der File-Requester

Ein File-Requester soll die Angabe von Datei- bzw. Pfadnamen erleichtern. Es findet der serienmäßig vorhandene File-Requester von OS 2.X Verwendung. Achten Sie darauf, daß bei Ladeoperationen der Hintergrund hell ist, während er bei Speichervorgängen dunkel erscheint! Das Suchmuster für die Dateinamen kann per ARexx-Befehl (SETPATTERN) verändert sowie ein- und ausgeschaltet werden. Die Einstellungen sämtlicher Requester können übrigens (zusammen mit anderen Parametern) komfortabel gesichert werden.

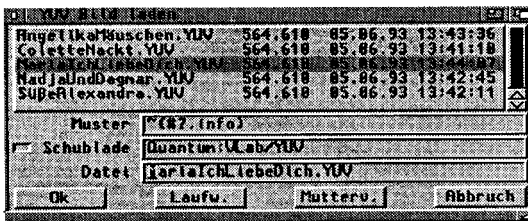


Abbildung 4.13: Der File-Requester von OS 2.X

#### 4.1.5 Der Font-Requester

Ein Font-Requester dient der Auswahl von Fonts für die Screentexte oder die Texte innerhalb von Fenstern. In dem Listview-Gadget links werden die zur Verfügung stehenden Schriften und im Listview-Gadget rechts die zur Verfügung stehenden Schriftgrößen angezeigt. Die Schriften werden durch einfaches Anklicken ausgewählt und sofort in dem Feld unter den beiden Listview-Gadgets angezeigt. Bei allen Schriften können im Integer-Gadget für die Schriftgröße auch beliebige Größen angegeben werden, woraufhin dieser Größe automatisch neu berechnet wird, was einen kleinen Augenblick dauern kann.

größere Menüs, klappen nicht mehr sichtbar herunter, sie lassen sich allerdings immer noch blind oder über Tastaturkürzel bedienen. War die Workbench geschlossen, kann sie nicht mehr geöffnet werden.

Abhilfe schafft hier nur die Freigabe von etwas Speicher. Neben Beenden anderer Programme oder Schließen bzw. Verkleinern unbenutzter Fenster bleibt natürlich die Möglichkeit, einen Bildspeicher von *VLab* zu löschen. Aber Vorsicht: Dabei gehen die zuvor eingelesenen Daten verloren! Mehr Speicher kann allerdings meistens durch Änderungen außerhalb von *VLab* freigemacht werden. Hier einige Beispiele:

- Alle (bis auf ein) Diskettenlaufwerke abschalten (danach Reset).
- Die Puffer der Diskettenlaufwerke und der Festplatte verkleinern.
- Kein „CPU FASTROM“ oder „SetCPU FASTROM“ benutzen.
- Keine Programme resident laden.
- Keine anderen Programme starten.
- Keinen anderen Font als Topaz-8 benutzen.
- Die Workbench auf zwei Farben und Non-Interlace umstellen.
- Das *VLab*-Makro `LoMem.vm` aufrufen.
- Evtl. über den Kauf einer Speichererweiterung nachdenken.

Ein wichtiger Punkt ist leider immer noch die angeblich hohe Absturz-Rate des Amiga, wenn der Speicher knapp wird. Dazu ein Beispiel: Ein Programm öffnet ein Fenster und gibt darin einen Text aus. Soweit, so gut. Was aber passiert, wenn sich das Fenster nicht öffnet? Das Amiga-Betriebssystem (Intuition) meldet das dem Programm, so daß entsprechende Maßnahmen (Fehlermeldung, Programmabbruch) getroffen werden können. Hat der Programmierer aber die Meldung ignoriert oder reagiert falsch, kommt es garantiert früher oder später (meist früher) zum Absturz, wenn der Text in das nicht vorhandene Fenster geschrieben wird. Leider ist es so, daß selbst bei umfangreichen und teuren Programmen bekannter Firmen häufig solche und ähnliche Fehler gemacht werden und der Schuldige dann der Amiga sein soll!

Bei *VLab* haben sich die Entwickler große Mühe gegeben, solche Fehler zu vermeiden. Natürlich kann es vorkommen, daß einige Funktionen bei akutem Speichermangel nicht mehr arbeiten, ein Absturz ist aber nicht zu erwarten! In der Regel, wenn dafür noch ausreichend Speicher frei ist, erfolgt eine Fehlermeldung.

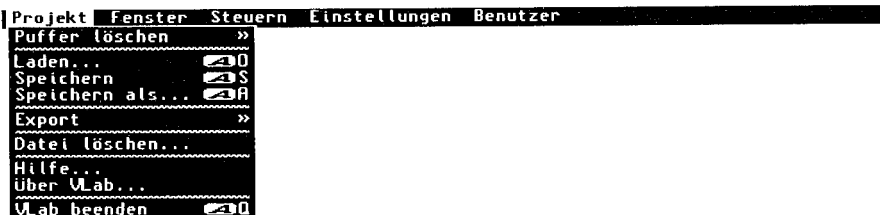
# Kapitel 5

## Die VLab-Funktionen im Detail

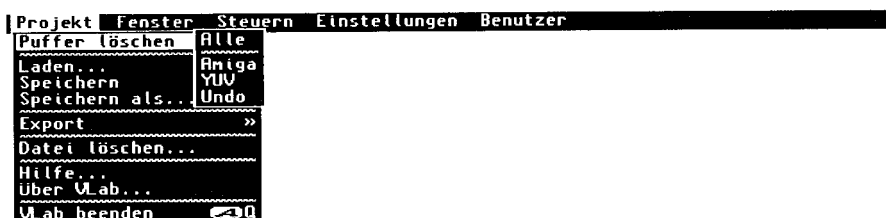
In den einzelnen *VLab*-Fenstern befinden sich jeweils eine Anzahl Gadgets, über welche die einzelnen Funktionen zu steuern sind. Jedes Fenster beinhaltet Gadgets, die sinngemäß zu einer Funktionsgruppe gehören. Im folgenden werden alle Menüpunkte, sämtliche Gadgets sowie die diesen Funktionen entsprechenden ARexx-Befehle erklärt. Außerdem werden einige ARexx-Befehle aufgeführt, die zwar keinen äquivalenten Menüpunkt oder ein entsprechendes Gadget besitzen, jedoch funktionell in enger Verbindung zu den erläuterten Menüs stehen. Diese Befehle können von einem ARexx-Programm aus angewendet werden oder direkt über Fenster » Befehl ausführen eingeben und ausgeführt werden. Eine Erläuterung der Schablonen für die ARexx-Befehle finden sie im Kapitel 8.

Bei verschiedenen Funktionen steht am Ende der Menüzeile ein eingerahmter Buchstabe. So wird die Tastaturabkürzung für das jeweilige Kommando dargestellt. Die Funktion kann also auch durch gleichzeitiges Drücken der rechten Amiga-Taste und der angegebenen Taste ausgelöst werden.

### 5.1 Menü: Projekt



## 5.1.1 Puffer löschen



### 5.1.1.1 Alle

Menü: **Projekt** » **Puffer löschen** » **Alle**

Gadget: —

ARexx: **CLEAR ALL**

Alle Bildspeicher innerhalb von *VLab* werden gelöscht und der Speicher freigegeben. Diese Funktion bietet sich an, wenn man kurzfristig den von *VLab* benötigten Speicher für die Ausführung eines anderen Programmes benötigt, *VLab* aber nicht vollständig verlassen will.

### 5.1.1.2 Amiga

Menü: **Projekt** » **Puffer löschen** » **Amiga**

Gadget: —

ARexx: **CLEAR AMIGA**

Hiermit kann der Speicher gelöscht werden, in dem sich das zuvor umgerechnete Amigabild befindet.

### 5.1.1.3 YUV

Menü: **Projekt** » **Puffer löschen** » **YUV**

Gadget: —

ARexx: **CLEAR YUV**

Es wird nur der Puffer für das zuletzt digitalisierte YUV-Bild geleert.

### 5.1.1.4 Undo

Menü: **Projekt** » **Puffer löschen** » **Undo**

Gadget: —

ARexx: **CLEAR UNDO**

Mit dieser Funktion wird der Undo-Puffer gelöscht und der belegte Speicher wieder freigegeben. Dieser Bildspeicher enthält in der Regel ein älteres YUV-Bild, das von Hand oder automatisch dort hineinkopiert wurde, um es nicht beim Einlesen eines neuen Bildes zu verlieren. Sollte der neue Schnappschuß mißlingen, kann dann auf das vorherige Bild im Undo-Puffer zurückgegriffen werden.

### 5.1.2 Laden

**Menü:** Projekt » Laden...

O

**Gadget:** —

**ARexx:** OPEN

Der File-Requester erscheint und erwartet die Auswahl eines zu ladenden YUV (Y)-Bildes. Es können sowohl die mit der Save-Funktion, als auch die mit der Sequenzfunktion abgespeicherten Bilder eingelesen werden.

### 5.1.3 Speichern

**Menü:** Projekt » Speichern

S

**Gadget:** —

**ARexx:** SAVE

Das im YUV-Speicher vorliegende Bild wird als YUV-IFF-Bild abgespeichert. Das Bild wird unter dem gleichen Namen abgelegt, der zuletzt beim Laden oder Speichern eines YUV-Bildes verwendet wurde. Dieser läßt sich ständig in der VLab-Titelleiste ablesen.

Steht die Anzahl der Sicherheitsabfragen auf selten oder häufig, erscheint eine Kontrollabfrage, bevor ein bereits existierendes Bild überschrieben wird.

Ist Sicherheitsabfragen auf keine gesetzt, wird ein evtl. bereits auf dem Datenträger vorhandenes Bild mit diesem Namen kommentarlos überschrieben!

### 5.1.4 Speichern als

**Menü:** Projekt » Speichern als...

A

**Gadget:** —

**ARexx:** SAVEAS

Hier wird das im YUV-Speicher abgelegte Bild ebenfalls abgespeichert, es erscheint aber in jedem Fall der File-Requester, um einen neuen Namen für das Bild zu erfragen.

## 5.1.5 Export



### 5.1.5.1 Amiga

Menü: Projekt » Export » Amiga...

X

Gadget: —

ARexx: EXPORTAMIGA

Die im Speicher für ein berechnetes Bild vorliegenden Daten werden als Amiga-IFF-Bild abgespeichert. Der Modus des Bildes richtet sich nach den im Umrechnen-Fenster eingestellten Parametern für Auflösung, Farbtiefe, Farbe oder Graustufen. Solange nicht einer der Spezialmodi mit mehr als 16 Graustufen oder mehr als 32 Farben (ausgenommen HAM) angewendet wurde, kann dieses Bild auf dem Amiga von fast jedem Programm, das mit IFF-Bildern arbeitet, eingelesen und direkt angezeigt werden. Die Funktion arbeitet natürlich nur, wenn schon ein berechnetes Bild vorliegt.

### 5.1.5.2 24 Bit ILBM

Menü: Projekt » Export » 24 Bit ILBM...

B

Gadget: —

ARexx: EXPORT24

Die Daten im YUV-Speicher werden als 24 Bit IFF-Bild abgespeichert. Dieser Modus empfiehlt sich dann, wenn das digitalisierte Bild in einem DTP-Programm oder einem Bildbearbeitungsprogramm weiterverarbeitet werden soll, das ausschließlich RGB-Bilder (mit 24 Bit-Farbtiefe) einlesen kann. Die Dateigröße für ein schirmfüllendes Bild steigt bei diesem Dateiformat von ca. 600 kB für das YUV-Bild auf ca. 1.1 MB für das 24 Bit RGB-Bild. Dabei werden die YUV-Daten interpoliert, um die volle Farbauflösung des RGB-Formats zu nutzen.

Leider lassen sich digitalisierte 24 Bit Bilddaten nur sehr schwierig komprimieren. Manchmal erhöht sich bei der Verwendung der einfachen IFF-Kompression sogar die Dateilänge, weshalb diese bei *VLab* per Menü ausschaltbar ist.

Wollen Sie beim Archivieren vieler *VLab*-Bilder Speicherplatz sparen, empfiehlt es sich, die Videobilder mit einem anderen Programm (wie z.B. ADPro2) in das JPEG-Format zu konvertieren. Der JPEG-Algorithmus ist ein ausgezeichnetes Verfahren zur Kompression von Standbildern. Zwar ist die Umrechnung in der Regel nicht völlig verlustfrei, die Stärke der Qualitätseinbuße ist aber einstellbar und bei geringer Einstellung bei digitalisierten Bildern mit dem bloßen Auge praktisch nicht wahrnehmbar. Dafür wird man mit einem Kompressionsverhältnis von 1:10 bis 1:40 belohnt, so daß ein 1,1 MB großes RGB-Bild auf ca. 100 bis runter zu 25 Kilobyte schrumpft.

### 5.1.5.3 24 Bit DEEP

**Menü:** Projekt » Export » 24 Bit DEEP...

V

**Gadget:** —

**ARexx:** EXPORT24 DEEP

Das aktuelle YUV-Bild wird in ein 24 Bit-RGB-Bild gewandelt und als IFF-DEEP-Datei abgespeichert. Dieses recht neue Format wurde von der Firma TecSoft für das Programm TVPaint erfunden und wird z.Zt. noch nicht von allen Amiga-Programmen zur Bildverarbeitung verstanden. Durch die im Vergleich zum IFF-ILBM-Format geänderte Reihenfolge, in der die einzelnen Bilddaten abgelegt werden, wird ein wesentlich schnelleres Laden und Speichern möglich.

### 5.1.6 Datei löschen

**Menü:** Projekt » Datei löschen...

**Gadget:** —

**ARexx:** DELETEFILE

Da es recht umständlich ist, immer auf die Workbench zu wechseln, wenn man mal wieder einige Dateien löschen muß, um Platz für neue *VLab*-Bilder zu schaffen, wurde diese Funktion gleich in *VLab* integriert. Das Kommando startet den bekannten File-Requester. Über das Listview-Gadget können Sie dann die zu löschende Datei auswählen.



### 5.1.7 Hilfe

**Menü:** Projekt » Hilfe...  
**Gadget:** —  
**ARexx:** HELP

Zeigt den Hilfetext an, der erläutert, wie durch Drücken der Help-Taste die Online-Hilfe für alle anderen Menüpunkte abgerufen werden kann.

### 5.1.8 Über VLab

**Menü:** Projekt » Über VLab...  
**Gadget:** —  
**ARexx:** ABOUT

Es erscheint ein Fenster, in dem neben der Versionsnummer von *VLab* alle Personen aufgeführt werden, die an der Entwicklung von *VLab* maßgeblich beteiligt waren. Weiterhin wird für den aktuellen VLab-Prozeß der Name des Public-Screens und des ARexx-Ports angezeigt.

### 5.1.9 VLab beenden

**Menü:** Projekt » VLab beenden Q  
**Gadget:** Close-Gadget des VLab-Screens oder -Fensters  
**ARexx:** QUIT

Sie wollen dieses wunderbare Programm schon wieder verlassen? Beim Beenden von *VLab* werden alle Einstellungen in einer Datei abgespeichert, sofern die Option Automatisch übernehmen bzw. Automatisch archivieren aktiviert ist. Beim nächsten Aufruf von *VLab* präsentiert es sich wieder so, wie Sie es verlassen haben. Sie können sofort weiterarbeiten, ohne Ihre Fenster und Optionen neu einrichten zu müssen.

## 5.2 Menü: Fenster



### 5.2.1 Einlesen

Menü: Fenster » Einlesen... 1

Gadget: —

ARexx: OPENWINDOW SCAN

Dieser Befehl öffnet das Fenster zur Steuerung des Digitalisiervorgangs. Die Bedienung der einzelnen Funktionen innerhalb dieses Fensters wird im Abschnitt 5.3.1 näher erläutert.

### 5.2.2 Umrechnen

Menü: Fenster » Umrechnen... 2

Gadget: —

ARexx: OPENWINDOW CONVERT

Mit diesem Menüpunkt öffnet man das Fenster zur Bedienung der Umrechnfunktionen. Die Benutzung der einzelnen Funktionen innerhalb dieses Fensters wird im Abschnitt 5.3.2 erklärt.

### 5.2.3 Sequenz

Menü: Fenster » Sequenz... 3

Gadget: —

ARexx: OPENWINDOW SEQUENCE

Durch dieses Kommando wird das Fenster für die Aufnahme einer Sequenz von *VLab*-Bildern geöffnet. Auf die Verwendung der einzelnen Gadgets innerhalb dieses Fensters wird im Abschnitt 5.3.3 näher eingegangen. Außerdem finden Sie in Abschnitt 6.5 (Seite 86) eine kurze Beschreibung des kompletten Vorgangs zur Aufnahme einer Sequenz.

## 5.2.4 Film (IFR)

Menü: Fenster » Film (IFR)...

4

Gadget: —

ARexx: OPENWINDOW IFR

Nach dem Aufruf dieses Menüeintrags öffnet sich das IFR-Fenster. Mit Hilfe der „Interleaved Frame Recording“ genannten Methode können Echtzeitsequenzen durch mehrfache, verschachtelte Aufnahmevorgänge erzeugt werden. Die Bedienung dieser speziellen Form der Sequenzaufnahme wird im Abschnitt 6.6 ab Seite 87 beschrieben.

## 5.2.5 Farbkorrektur

Menü: Fenster » Farbkorrektur...

5

Gadget: —

ARexx: OPENWINDOW COLOR, COLORVALUE

Mit diesem Menüeintrag wird das Fenster für die Farbkorrektur geöffnet.

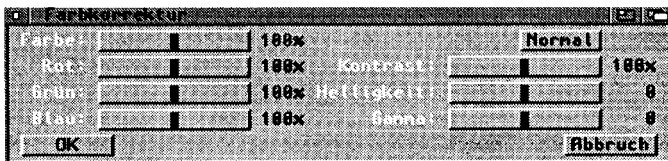


Abbildung 5.1: Das Farbkorrektur-Fenster

Im Gegensatz zu vielen anderen Videodigitizern besitzt die *VLab*-Hardware keine Regler um z.B. die Bildhelligkeit oder die Farben zu beeinflussen. Eine ausgeklügelte Automatik sorgt für eine gleichbleibend gute Bildqualität auch bei wechselnden Signalquellen. Solange das Signal am FBAS-Eingang von *VLab* eine gute Qualität hat (was bei Aufnahmen direkt vom Fernsehen oder einem Bildplattenspieler in der Regel der Fall ist), ist eine solche Veränderung des Signals auch nicht notwendig.

Bei Aufnahmen von preiswerteren Videorecordern oder direkt von einer Kamera ist eine Veränderung des Bildes aber häufig durchaus sinnvoll. Auch dabei sorgt die Automatik natürlich dafür, daß das Bild möglichst unverfälscht digitalisiert wird. Mit Hilfe der im folgenden beschriebenen Möglichkeiten kann die Bildqualität während einer Konvertierung des digitalisierten YUV-Bildes in ein Amiga-Format oder in 24 Bit beeinflusst werden. Bei vorsichtiger Anwendung der Slider, d.h. wenn sie nur wenig von ihrer Normalstellung wegbewegt werden, ist häufig eine zumindest subjektive Verbesserung des Bildes möglich. Werden einige Slider zu weit von ihrer Normalstellung wegbewegt, wird das Bild selbstverständlich verfremdet — aber auch das ist teilweise durchaus erwünscht.

### 5.2.5.1 Farbe

**Menü:** —

**Gadget:** **Slider Farbe im Fenster Farbkorrektur**

**ARexx:** **COLORVALUE CHROMINANCE 0... 200**

Mit diesem Slider kann die Farbsättigung (Chrominanz) zwischen null (schwarzweiß) und 200 Prozent (sehr bunt) eingestellt werden. Dadurch können farbschwache Aufnahmen oder Bilder mit zu grellen Farben korrigiert werden. Die Helligkeit (Luminanz) wird nicht verändert.

### 5.2.5.2 Rot, Grün, Blau

**Menü:** —

**Gadget:** **Slider Rot, Grün, Blau im Fenster Farbkorrektur**

**ARexx:** **COLORVALUE RED | GREEN | BLUE 0... 200**

Diese drei Regler beeinflussen die Intensität der drei additiven Grundfarben. Damit lassen sich Farbstiche (etwa bei falschem Weißabgleich) beheben oder besondere Effekte erzielen.

### 5.2.5.3 Kontrast

**Menü:** —

**Gadget:** **Slider Kontrast im Fenster Farbkorrektur**

**ARexx:** **COLORVALUE CONTRAST 0... 200**

Durch diesen Schieber wird der Kontrast, also der Gegensatz zwischen hell und dunkel, erhöht oder verringert.

### 5.2.5.4 Helligkeit

**Menü:** —

**Gadget:** **Slider Helligkeit im Fenster Farbkorrektur**

**ARexx:** **COLORVALUE LUMINANCE -100... 100**

Die Grundhelligkeit für alle Farben wird durch diesen Regler verändert.

### 5.2.5.5 Gamma

**Menü:** —

**Gadget:** **Slider Gamma im Fenster Farbkorrektur**

**ARexx:** **COLORVALUE GAMMA -100... 100**

Bei der Gamma-Korrektur wird durch eine besondere Formel der Kontrast so beeinflusst, daß keine negativen oder zu großen Werte entstehen können, die sonst abgeschnitten werden müßten. Man kann sie z.B. zur Korrektur von Belichtungen einsetzen. Positive Einstellungen hellen auf, während negative Werte zur Abdunkelung führen.

#### 5.2.5.6 Normal

**Menü:** —

**Gadget:** **Button Normal im Fenster** Farbkorrektur

**ARexx:** TRUECOLOR

Nach Betätigen dieses Gadgets bzw. Senden des ARexx-Befehls werden alle Einstellungen zur Farbkorrektur auf ihre Normalwerte gebracht. Ein danach konvertiertes Bild wird bei der Umrechnung in ein Amiga- oder 24 Bit-Bild nicht mehr verändert. Besonders in ARexx-Scripts sollte TRUECOLOR vor jeder Konvertierung zur Sicherheit aufgerufen werden.

#### 5.2.5.7 OK

**Menü:** —

**Gadget:** **Button OK im Fenster** Farbkorrektur

**ARexx:** —

Das Fenster zum Einstellen der Farbkorrektur wird geschlossen. Die Einstellungen der Schieberegler werden übernommen und bei allen folgenden Umrechnungen benutzt.

#### 5.2.5.8 Abbruch

**Menü:** —

**Gadget:** **Button Abbruch im Fenster** Farbkorrektur

**ARexx:** —

Das Fenster zum Einstellen der Farbkorrektur wird geschlossen. Die zuvor gemachten Veränderungen an den Slidern werden rückgängig gemacht. Bei den folgenden Bildumrechnungen werden wieder die Farbkorrekturwerte benutzt, die vor dem letzten Öffnen des Fensters gültig waren.

Hinweis: Eine ausführlichere Beschreibung der *VLab*-Farkorrektur befindet sich in Abschnitt 6.7 auf Seite 96.

## 5.2.6 Quelle wählen

6

**Menü:** Fenster » Quelle wählen...

**Gadget:** —

**ARexx:** OPENWINDOW SELECTSOURCE

Hiermit wird ein Fenster für die Auswahl einer Quellendefinition geöffnet.

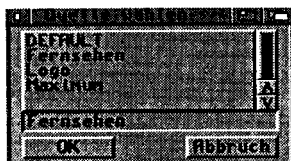


Abbildung 5.2: Das Fenster zur Auswahl einer **Quellendefinition**

### 5.2.6.1 Auswahl

**Menü:** —

**Gadget:** Listview im Fenster Quelle wählen

**ARexx:** SELECTSOURCE Quellename

Über dieses Listview-Gadget kann eine zuvor definierte Signalquelle ausgesucht werden. Der angeklickte Videoquellen-Parametersatz wird dann sofort für den Monitorbetrieb und beim Einlesen verwendet. Für eine schnelle Auswahl kann die gewünschte Quellendefinition mit einem Doppelklick gewählt werden; das Fenster wird daraufhin geschlossen.

Zu einer Quellen-Definition gehören viele verschiedene Faktoren, die in dem Abschnitt „Quelle definieren“ einzeln beschrieben sind.

### 5.2.6.2 OK

**Menü:** —

**Gadget:** Button OK im Fenster Quelle wählen

**ARexx:** SELECTSOURCE Quellename

Wurde die Quellendefinition nicht durch einen Doppelklick ausgewählt, kann dies durch Anklicken des Button-Gadgets OK geschehen.

### 5.2.6.3 Abbruch

Menü: —

Gadget: **Button** Abbruch im Fenster Quelle wählen

ARexx: —

Wurde die Quellendefinition nicht durch einen Doppelklick gewählt, kann die Auswahl durch Anklicken des Button-Gadgets Abbruch abgebrochen werden. Es wird dann die zuvor benutzte Quellendefinition weiter verwendet.

### 5.2.7 Quelle definieren

Menü: Fenster  $\gg$  Quelle definieren...

7

Gadget: —

ARexx: OPENWINDOW DEFINESOURCE

Öffnen des Fensters für die Definition einer Videoquelle.

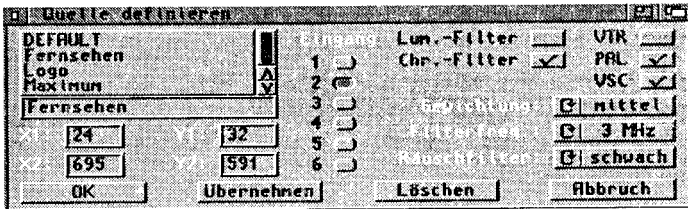


Abbildung 5.3: Das Fenster für die **Definition einer Quelle**

#### 5.2.7.1 Quelle

Menü: —

Gadget: **Listview und String** im Fenster Quelle definieren

ARexx: SELECTSOURCE Quellename

Jede Signalquellen-Definition wird unter einem eigenen Namen abgelegt. Ein Mausklick auf den Namen einer gewünschten Quelle im Listview-Gadget erklärt diesen Videoquellen-Parametersatz zum gerade aktuellen. Alle anderen Gadgets in diesem Fenster werden dann auf die Werte dieser Quellendefinition umgestellt, denn ihre Einstellungen sind bei jeder definierten Quelle einzeln gespeichert. Bei der Neudefinition einer Signalquelle muß man einfach einen neuen Namen in das String-Gadget unter dem Listview-Gadget eingetragen. Danach wird man alle anderen Gadgets auf die gewünschte Einstellung bringen. Ein Klick auf das Gadget Übernehmen oder OK übernimmt die neue Definition in die Quellenliste. *VLab* merkt sich nun den neuen Parametersatz.

### 5.2.7.2 Bildausschnitt

**Menü:** —

**Gadget:** Integer X1, Y1, X2, Y2 im Fenster Quelle definieren

**ARexx:** DEFINECLIP X1 Y1 X2 Y2

Die Werte in den vier Integer-Gadgets X1, Y1 und X2, Y2 geben die linke obere und die rechte untere Ecke des gewünschten Bildausschnittes an. Die Angabe 0, 0 und 719, 624 stellt das absolute Maximum dar. Das entspricht einer maximalen horizontalen Auflösung des VLab-Digitizers von 720 Bildpunkten und der vertikalen Auflösung des PAL-Videostandards von 625 Zeilen. Da übliche Videoquellen diesen Bereich aber nicht voll ausnutzen — beim Fernsehen werden in den ersten Zeilen die Videotextinformationen übertragen, bei Videokameras existiert ein schwarzer Rand um das eigentliche Bild — genügt es in der Regel, einen kleineren Bildausschnitt zu wählen, etwa 24, 32 und 695, 591. Diese Einstellung entspricht einem Bild mit 672×560 Bildpunkten. Wird die Eingabe in einem der Gadgets mit Return beendet, spätestens jedoch am Ende einer Quellen-Definition, werden die Werte überprüft. Ergeben die Werte keinen Sinn oder sind sie außerhalb des zulässigen Bereiches, erfolgt eine entsprechende Meldung.

### 5.2.7.3 Eingang

**Menü:** Steuern » Videoeingang » 1, 2, 3, 4, 5, 6

**Gadget:** MX Eingang im Fenster Quelle definieren

**ARexx:** INPUT 1..6

Jede VLab-Hardware besitzt zwei bis drei Videobuchsen. Über das MX-Gadget Eingang kann eine davon angewählt werden. Anschluß Nummer drei (bzw. sechs) ist für die SVHS-Buchse vorgesehen. Die höheren Eingänge (4..6) können nur dann angesprochen werden, wenn ein weiteres VLab-Gerät im Rechner eingebunden ist. Die Belegung der Videoanschlüsse für die unterschiedlichen VLab-Modelle ist Abschnitt 3.4 zu entnehmen.

### 5.2.7.4 VTR

**Menü:** Steuern » Hardware » Videorecorder

**Gadget:** Checkbox VTR im Fenster Quelle definieren

**ARexx:** VTRMODE ON | OFF

Viele Echtzeit-Digitizer haben Probleme, wenn sie an einen Videorecorder angeschlossen werden. Der Grund besteht darin, daß bei der Wiedergabe vom Band leichte Gleichlaufschwankungen auftreten, so daß die Bildwiederholfrequenz immer leicht von 50 Hz (bei PAL-Systemen) abweicht. In der professionellen Videotechnik wird in solchen Fällen ein *Time-Base-Corrector* eingesetzt, um das Videosignal des Recorders mit einem Studiotakt



zu synchronisieren. Diese Geräte sind allerdings recht teuer. Mit der *VLab*-Hardware tritt dieses Problem nicht auf, da sie eine eingebaute Time-Base-Correction besitzt, die über dieses Gadget aktiviert werden kann. Das VTR-Gadget sollte bei allen Quellendefinitionen für Videorecorder aktiviert werden. Wurde die Time-Base-Correction nicht eingeschaltet, kann von einem Videorecorder meist nur ein Halbbild ohne Schwierigkeiten aufgenommen werden. Beim Vollbild zeigt sich meist ein stark verschobenes zweites Halbbild, wie bei einer sehr schnellen Bewegung.

#### 5.2.7.5 PAL

**Menü:** —

**Gadget:** **Checkbox PAL im Fenster** Quelle definieren

**ARexx:** PALMODE ON | OFF

*VLab* gibt es nicht in zwei verschiedenen Versionen für PAL und NTSC, da die *VLab*-Hardware problemlos nach beiden Verfahren kodierte Farbsignale verarbeiten kann. Die Umschaltung erfolgt einfach über dieses Gadget. Auf eine automatische Umschaltung wurde bewußt verzichtet, weil diese nur bei Farbbildern zuverlässig arbeiten könnte, bei Schwarzweiß-Signalen müßte ohnehin manuell umgeschaltet werden. Bei einer NTSC-Quelle sinkt die maximal einstellbare (digitalisierbare) Zeilenzahl auf 525 Zeilen.

#### 5.2.7.6 VSC

**Menü:** —

**Gadget:** **Checkbox VSC im Fenster** Quelle definieren

**ARexx:** VSYNCCORRECTION ON | OFF

VSC steht für *Vertical Sync Correction*. Bei schlechten Signalquellen, z.B. verrauschtem oder mit Geisterbildern durchsetztem Fernsehempfang kann es vorkommen, daß das Bild vertikal um einige Zeilen springt. Das ist dann besonders bei laufendem Monitorbild zu erkennen. Wird VSC eingeschaltet, steht das Bild wieder ruhig. Beim Wechsel der Signalquelle, z.B. nach Umschalten auf ein anderes Fernsehprogramm, kann es bei der erfolgreichen Neusynchronisation dann aber vorkommen, daß das Bild falsch „einrastet“. Tritt ein solcher Effekt auf, sollte die Hardware durch einen *VLab*-Reset zurückgesetzt werden.

#### 5.2.7.7 Luminanz-Filter

**Menü:** —

**Gadget:** **Checkbox Lum.-Filter im Fenster** Quelle definieren

**ARexx:** LUMINANCEFILTER ON | OFF

Hiermit kann auf der Hardware ein spezieller Luminanz-Filter ein- oder ausgeschaltet werden. Die Luminanz-Information ist die Helligkeitsamplitude des Videosignals. Zum Digitalisieren von Farbbildern sollte er abgeschaltet werden — dann muß nämlich ausschließlich der Chrominanz-Filter aktiviert sein. Beim Einlesen von z.B. einer Schwarzweiß-Kamera, muß der Luminanz-Filter in der Regel eingeschaltet werden.

### 5.2.7.8 Chrominanz-Filter

**Menü:** —

**Gadget:** **Checkbox** Chr.-Filter **im Fenster** Quelle definieren

**ARexx:** **CHROMAFILTER ON | OFF**

Der Chrominanzfilter muß beim Digitalisieren aus farbigen Videoquellen unbedingt eingeschaltet sein, selbst wenn diese gerade nur S/W-Bilder wiedergeben oder nur in Schwarzweiß eingelesen werden soll. Der Chrominanzfilter trennt die Farbinformation des Videosignals ab, bevor die Helligkeitsinformation digitalisiert wird. Ist der Filter nicht eingeschaltet, beeinflusst die Farbinformation das Helligkeitssignal, was zu einer sehr schlechten Bildqualität führt: Das Bild wird sehr „grieselig“. Nur bei reinen Schwarzweiß-Signalquellen (z.B. S/W-Kamera) kann er abgeschaltet werden, wodurch die horizontale Auflösung und die Bildschärfe erhöht werden kann.

### 5.2.7.9 Gewichtung

**Menü:** —

**Gadget:** **Cycle** Gewichtung **im Fenster** Quelle definieren

**ARexx:** **FILTERWEIGHT OFF | LIGHT | MEDIUM | STRONG**

Über die Gewichtung wird eingestellt, wie stark sich die beiden Funktionen Filterfrequenz und Rauschfilter auf das digitalisierte Videobild auswirken sollen. In Stellung aus haben die Werte von Filterfrequenz und Rauschfilter keinen Einfluß. Die beste Einstellung für Gewichtung und Filterfrequenz hängt immer von dem individuellen Bild ab, das Sie Digitalisieren wollen, und auf die Wiedergabe welcher Details Sie dabei besonderen Wert legen. Bei Aktivierung der Gewichtung setzt meist eine Konturanhebung in einigen Bereichen des Videobildes ein. Bei zu starker Gewichtung können diese Details aber verschwinden.

Für Landschaftsbilder und ähnliches empfiehlt sich eine schwächere Gewichtung, bei Zeichnungen, die erst durch einen hohen Kontrast richtig zur Geltung kommen, sollte die Gewichtung stärker gewählt werden. Dabei sollte dann die Filterfrequenz auf die Auflösung der Zeichnung und der Kamera abgestimmt werden.

Die hier vorgeschlagenen Parameterwerte können nur als Anhaltspunkt für Ihre eigenen persönlichen Einstellungen dienen. Experimentieren Sie ein wenig mit den vorhandenen Möglichkeiten, um die für Ihre jeweilige Anwendung optimalen Einstellungen zu finden, welche immer auch von dem jeweilig vorliegenden Videobild abhängig sein werden.

### 5.2.7.10 Filterfrequenz

**Menü:** —

**Gadget:** Cycle Filterfreq. im Fenster Quelle definieren

**ARexx:** FILTERFREQUENCY 2.5 | 3 | 3.5 | 4

Hiermit kann die Frequenz eines Bandpaßfilters eingestellt werden. Dieser Filter bewirkt eine Kontrastanhebung mancher Konturen. Durch geschickte Einstellung der Gewichtung und der Filterfrequenz läßt sich, je nach Güte der Signalquelle, die Qualität der eingelesenen Videobilder beeinflussen. Am besten sind die Auswirkungen an einem Testbild zu überprüfen.

Grob gesagt, ist die Filterfrequenz ein Maß für die Feinheit der Konturen, die dadurch hervorgehoben werden. Bei einer hohen Frequenz, vor allem wenn direkt von einer guten Kamera digitalisiert wird, werden vor allem feine Linien, z.B. bei Zeichnungen, betont. Bei hohen Frequenzen nimmt die Wirkung des Filters in Bezug auf die Konturanhebung ab. Bei niedrigen Frequenzen wirken die betonten Konturen sehr „grob-schlächtig“.

### 5.2.7.11 Rauschfilter

**Menü:** —

**Gadget:** Cycle Rauschfilter im Fenster Quelle definieren

**ARexx:** NOISEFILTER OFF | LIGHT | MEDIUM | STRONG

Die Wirkung dieses Rauschfilters sollte nicht überschätzt werden. Ein Bild, daß man schon auf größere Entfernung als „verschneit“ erkennen kann, läßt sich damit nicht wesentlich verbessern. Allerdings kann die Qualität von Bildern, die nur bei genauerem Hinsehen etwas Schnee zeigen, weiter verfeinert werden. Auch hier ist die Gewichtung von entscheidender Bedeutung!

### 5.2.7.12 OK

**Menü:** —

**Gadget:** Button OK im Fenster Quelle definieren

**ARexx:** ADDSOURCE

Wird OK gewählt, werden alle zuvor gemachten Einstellung in diesem Fenster mit dem Namen der Signalquelle in die Liste aufgenommen. Vorher erfolgt noch eine Überprüfung der eingestellten Werte. Ist alles in Ordnung, wird anschließend das Fenster geschlossen. Soll das Fenster geöffnet bleiben, ist statt OK Übernehmen zu benutzen.

### 5.2.7.13 Übernehmen

**Menü:** —

**Gadget:** **Button Übernehmen im Fenster** Quelle definieren

**ARexx:** **ADDSOURCE**

Wie bei OK werden bei Wahl von Übernehmen die zuvor gemachten Einstellungen nach Überprüfung in die Liste übernommen. Das Fenster bleibt jedoch offen, sodaß direkt weitere Eingaben gemacht werden können.

### 5.2.7.14 Löschen

**Menü:** —

**Gadget:** **Button Löschen im Fenster** Quelle definieren

**ARexx:** **DELETESOURCE Quellename**

Hiermit kann eine Quellen-Definition gelöscht werden, falls sie nicht mehr benötigt wird. Die zu löschende Quelle wird zunächst über das Listview-Gadget anhand ihres Namens angewählt. Danach steht der Name in dem darunterliegenden String-Gadget und bei Betätigung von Löschen wird der Eintrag entfernt. Die Löschung kann durch Abbruch nicht rückgängig gemacht werden. Man kann jedoch die Definition zur Not durch Neuladen von zuvor gespeicherten Einstellungen zurückholen.

### 5.2.7.15 Abbruch

**Menü:** —

**Gadget:** **Button Abbruch im Fenster** Quelle definieren

**ARexx:** —

Das Fenster wird geschlossen und alle vorgenommenen Änderungen an der aktuellen (!) Quellen-Definition werden rückgängig gemacht. Definitionen, die in die Liste übernommen oder aus ihr gelöscht wurden, werden allerdings nicht beeinflusst!

## 5.2.8 Befehl ausführen

**Menü:** **Fenster** >> **Befehl ausführen...**

9

**Gadget:** —

**ARexx:** **OPENWINDOW COMMAND**

Öffnen des Fensters für die Eingabe eines *VLab*-ARexx-Befehls.

Hinweis: Durch das mitgelieferte Makro „VShell.rx“ können auch ganze Folgen von *VLab*-ARexx-Kommandos komfortabel hintereinander eingetippt werden.

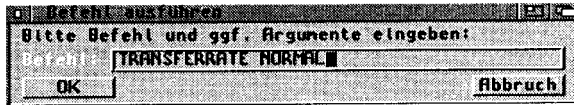


Abbildung 5.4: Das Fenster zur **Befehlseingabe**

### 5.2.8.1 Befehl

**Menü:** —

**Gadget:** **String** Befehl im Fenster Befehl ausführen

**ARexx:** — (nur Befehlsname und Parameter eingeben)

Alle ARexx-Befehle, die *VLab* versteht, können hier auch direkt eingegeben werden. Praktisch jeder Befehl in *VLab* besitzt ein entsprechendes ARexx-Äquivalent. Der auszuführende Befehl wird mit seinen Parametern eingetippt. Wird danach die Taste Return bzw. Eingabe gedrückt oder das Button-Gadget OK gewählt, wird der eingegebene Befehl ausgeführt. Das Fenster wird erst nach seiner Beendigung geschlossen.

### 5.2.8.2 OK

**Menü:** —

**Gadget:** **Button** OK im Fenster Befehl ausführen

**ARexx:** —

Der zuvor eingetippte Befehl wird ausgeführt.

### 5.2.8.3 Abbruch

**Menü:** —

**Gadget:** **Button** Abbruch im Fenster Befehl ausführen

**ARexx:** —

Das Fenster Befehl ausführen wird geschlossen, ohne daß das eingegebene Kommando ausgeführt wird.

## 5.2.9 Fenster schließen

0

**Menü:** Fenster » Fenster schließen...

**Gadget:** Close-Gadget des zu schließenden Fensters

**ARexx:** CLOSEWINDOW Fenstername

Das ausgewählte Fenster wird geschlossen. Mit dem Tastaturkürzel kann man, neben dem gerade aktiven Fenster, auch den Vorschau- oder Anzeige-Screen per Tastendruck schließen.

## 5.3 Menü: Steuern



### 5.3.1 Einlesen

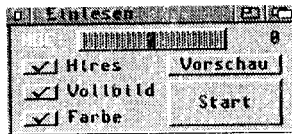
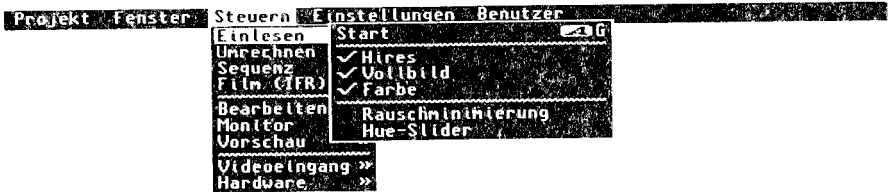


Abbildung 5.5: Das **Einlesen**-Fenster

#### 5.3.1.1 Start

Menü: **Steuern** » **Einlesen** » **Start**

G

Gadget: **Button Start im Fenster Einlesen**

ARexx: **SCAN**

Dieser Befehl startet die Digitalisierung eines Bildes in Echtzeit und das Übertragen des Bildes aus dem Bildspeicher der Digitalisierkarte in den Systemspeicher des Amiga. Das Bild wird in Abhängigkeit der Einstellungen für Vollbild/Halbbild, HiRes/LoRes und Schwarzweiß/Farbe in den YUV-Puffer von *VLab* eingelesen. Von dort kann es dann angezeigt, umgerechnet oder gespeichert werden. Ein y oder yuv in der Titelleiste zeigt die Belegung des YUV-Puffers an.

### 5.3.1.2 HiRes

**Menü:** **Steuern** » **Einlesen** » **HiRes**

**Gadget:** **Checkbox HiRes im Fenster Einlesen**

**ARexx:** **SCANMODE HIRES | LORES**

Diese Einstellung entscheidet darüber, ob beim Einlesen ein HiRes oder ein LoRes Bild im YUV-Speicher abgelegt wird. Ein HiRes-Bild besitzt eine maximale horizontale Auslösung von 720 Punkten, ein LoRes-Bild maximal 360 Punkte.

Das Einlesen in LoRes benötigt kurzzeitig etwas zusätzlichen Speicher und etwas mehr Zeit, da das Bild in diesem Fall beim Einlesen direkt von HiRes nach LoRes umgerechnet wird.

### 5.3.1.3 Vollbild

**Menü:** **Steuern** » **Einlesen** » **Vollbild**

**Gadget:** **Checkbox Vollbild im Fenster Einlesen**

**ARexx:** **SCANMODE LACE | NONLACE**

Ist Vollbild ausgewählt, wird beim nächsten Digitalisieren ein Vollbild statt eines Halbbildes eingelesen. Das Vollbild besitzt die doppelte vertikale Auflösung des Halbbildes, braucht dafür aber auch den doppelten Speicherplatz. Desweiteren benötigen bei einem Vollbild alle bildbearbeitenden Funktionen wie Vorschau, Umrechnen, Speichern die doppelte Rechenzeit. Sollten Sie einen normalen Amiga ohne Turbokarte besitzen, empfiehlt es sich, die meiste Zeit mit Halbbildern zu arbeiten und erst für das endgültige Bild (wenn die maximale Auflösung und Qualität erforderlich wird) die Vollbildfunktion einzuschalten.

Beim Einlesen von Vollbildern können Effekte auftreten, welche die Bildqualität zu senken scheinen. Die Entstehung dieser Effekte, ihre Vermeidung und die Anwendung der DeInterlace-Funktion werden in Abschnitt 6.2 näher beschrieben.

### 5.3.1.4 Farbe

**Menü:** **Steuern** » **Einlesen** » **Farbe**

**Gadget:** **Checkbox Farbe im Fenster Einlesen**

**ARexx:** **SCANMODE BW | COLOR**

Nur wenn der Menüpunkt Farbe angewählt ist, wird beim nächsten Digitalisiervorgang ein Farbbild eingelesen. Andernfalls wird lediglich ein Schwarzweißbild digitalisiert, also nur die Luminanzinformation (Y) des Videosignals eingelesen, was etwas schneller geht. Wenn ein Schwarzweißbild eingelesen wurde, wird in der Menüzeile nur y angezeigt, bei einem Farbbild yuv.



### 5.3.1.5 Rauschminimierung

**Menü:** Steuern » Einlesen » Rauschminimierung

**Gadget:** —

**ARexx:** MINIMIZE NOISE ON | OFF | LEVEL Mittelungen

Schaltet der Anwender diese Sonderfunktion ein, dann werden bei jedem Einlesevorgang mehrere Videobilder direkt hintereinander digitalisiert und zu einem Gesamtbild zusammengemischt. Dadurch kann man bei Standbildern zufällig verteilte Bildfehler (Rauschen) herausmitteln lassen. So können etwa bei Videokamera-Aufnahmen mit schwacher Ausleuchtung erhebliche Verbesserungen der Bildqualität erzielt werden.

Standardmäßig wird die Mittelung über vier Bilder ausgeführt. Die Anzahl der einzulesenden Einzelbilder kann aber mit dem LEVEL-Parameter des ARexx-Befehls MINIMIZE NOISE verändert werden.

### 5.3.1.6 Hue-Slider

**Menü:** Steuern » Einlesen » Hue-Slider

**Gadget:** Slider im Fenster Einlesen ein-/ausschalten

**ARexx:** HUESLIDER ON | OFF

Nach Aktivierung dieser Option erscheint im Einlesen-Fenster ein Schieberegler, mit dem bei NTSC-Betrieb auftretende Farbstiche korrigiert werden können. Bei Videoquellen in PAL-Norm darf die Hue-Funktion nicht verwendet werden.

### 5.3.1.7 Slowscan

**Menü:** —

**Gadget:** —

**ARexx:** SLOWSCAN AUTO | ON | OFF

Auf einem Amiga mit nur 1 MB RAM ohne Speichererweiterung oder Turbo-Karte wird der Amiga bei der Darstellung eines 16-Graustufen HiRes-Screens so langsam, daß kein einwandfreies Digitalisieren und Einlesen mehr möglich ist. Für diesen Fall kann der sogenannte Slowscan-Modus gewählt werden, der dann während der zeitkritischen Operationen kurzzeitig die Bildschirmausgabe abschaltet. Normalerweise wird die Notwendigkeit dazu vom Programm automatisch erkannt. Da es jedoch auch Fälle geben könnte, in denen die Analyse fehlschlägt, kann die Automatik abgeschaltet werden.

Die Funktion ist nur als ARexx-Befehl vorhanden, kann aber außerdem in der Datei „VLab.Startup“, in einem Makro oder direkt als Befehl benutzt werden.

## 5.3.2 Umrechnen

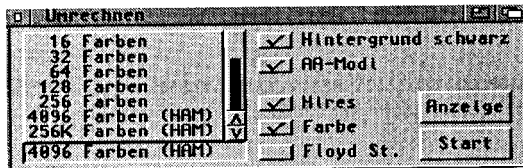


Abbildung 5.6: Das Umrechnen-Fenster

### 5.3.2.1 Start

Menü: **Steuern >> Umrechnen >> Start** T

Gadget: **Button Start im Fenster Umrechnen**

ARexx: **CONVERT**

Die Umrechnung in den zuvor gewählten Bildtyp wird gestartet. Sie kann je nach Typ und Rechenleistung des benutzten Amiga einige Sekunden bis zu mehreren Minuten dauern, was aber im Vergleich mit anderen Programmen immer noch recht schnell ist. Ist das Bild fertig berechnet, kann es entweder über das Button-Gadget Anzeige oder auch automatisch dargestellt werden (s.u.). Ein Abspeichern des Bildes im IFF-Format (Export Amiga) ist dann ebenfalls möglich.

### 5.3.2.2 Bild anzeigen

Menü: **Steuern >> Umrechnen >> Bild anzeigen** D

Gadget: **Button Anzeige im Fenster Umrechnen**

ARexx: **DISPLAY**

Ist ein Bild in einem Amiga-Format vorhanden (AP in der Titelleiste), kann es hiermit angezeigt werden. Auf einem separaten Screen wird das Bild in dem Modus und der Größe dargestellt, wie es nach der Konvertierung im Speicher vorliegt.

### 5.3.2.3 Bild automatisch anzeigen

**Menü:** Steuern » Umrechnen » Anzeige autom.

**Gadget:** —

**ARexx:** AUTODISPLAY ON | OFF

Diese Option legt fest, ob das konvertierte Bild nach Beendigung der Umrechnung automatisch angezeigt wird oder nicht.

### 5.3.2.4 HiRes

**Menü:** Steuern » Umrechnen » Hires

**Gadget:** Checkbox Hires im Fenster Umrechnen

**ARexx:** CONVERTMODE LORES | HIRES

Liegt vor der Umrechnung in ein Amiga-Bild ein HiRes-Bild vor und wurde HiRes gewählt, ist auch die erzeugte Amiga-Grafik wieder ein HiRes-Bild. Ist HiRes nicht aktiv, erfolgt bei der Konvertierung gleichzeitig eine Skalierung auf LoRes.

Liegt das digitalisierte Bild bereits als LoRes-Bild vor, ist allerdings keine Skalierung auf HiRes möglich, es erfolgt dann eine entsprechende Fehlermeldung. In diesem Fall kann nur ein LoRes-Bild berechnet werden.

### 5.3.2.5 AA-Modi

**Menü:** Steuern » Umrechnen » AA-Modi

**Gadget:** Checkbox AA-Modi im Fenster Umrechnen

**ARexx:** SPECIALCONVERT ON | OFF

*VLab* kann die eingelesenen YUV-Bilder auch in die Grafikmodi der neuen Amiga-Modelle (Amiga 4000 und 1200) umrechnen. Da diese Bilder auf einem älteren Gerät jedoch nicht dargestellt werden können, werden diese Farbtiefen nur dann in dem Listview-Gadget angezeigt, wenn die Anzeige der „AA-Modi“ aktiviert wurde. Die entstehenden Grafiken können auf solchen Rechnern zwar nicht angezeigt, wohl aber als IFF-Bild zur Weiterverarbeitung mit DTP-Programmen oder anderen Computern (auch IBM-PCs) abgespeichert werden.

### 5.3.2.6 Farbe

**Menü:** Steuern » Umrechnen » Schwarzweiß, Farbe, HAM

**Gadget:** Checkbox Farbe im Fenster Umrechnen

**ARexx:** CONVERTMODE BW | COLOR | HAM

Vor der Umwandlung in ein Amiga-Bild muß die gewünschte Farbauflösung des Amiga-Bildes ausgesucht werden.

Ist Schwarzweiß gewählt, wird das YUV-Bild in ein Schwarzweißbild umgerechnet. Dabei muß vor dem Start der Konvertierung noch die gewünschte Anzahl Graustufen eingestellt werden. Der Amiga kann maximal 16 Graustufen darstellen, unabhängig von der gewählten Auflösung.

Bei der Umrechnung in Farbe muß die gewünschte Farbanzahl gewählt werden. Im LoRes-Modus können dabei bis zu 32 Farben verwendet werden. Ist der HiRes-Modus aktiviert, sind maximal 16 Farben zulässig.

Zusätzlich gibt es im LoRes-Modus noch die Möglichkeit, das YUV-Bild in ein HAM (Hold and Modify)-Bild mit bis zu 4096 Farben umzuwandeln. Dabei treten bei einem HAM-Bild die HAM-typischen Farbverläufe zwischen starken Farbkontrasten auf. Die Software versucht zwar beim Umrechnen das Auftreten dieser Farbverläufe durch geschickte Wahl der 16 Grundfarben zu minimieren, es kann aber bei einigen Bildern nicht vollständig vermieden werden, so daß dort eine Verringerung der horizontalen Auflösung des Amigabildes in Kauf genommen werden muß. Trotz dieses Nachteils ergeben sich in diesem Modus in der Regel die originalgetreuesten Amiga-Grafiken.

Beim Amiga 4000 kann auch das verbesserte HAM 8 mit bis zu 262144 Farbtönen verwendet werden. Zusätzlich läßt sich das 4096-Farben-HAM in HiRes nutzen. Außerdem ist es möglich, die Farbmodi mit bis zu 256 Farben (oder Graustufen) auf dem Amiga 4000 darzustellen.

### 5.3.2.7 Floyd-Steinberg

**Menü:** Steuern » Umrechnen » Floyd Steinberg

**Gadget:** Checkbox Floyd-St. im Fenster Umrechnen

**ARexx:** DITHER ON | OFF

Da der Amiga leider nur recht wenig Farben oder Graustufen gleichzeitig anzeigen kann (Ausnahme HAM), bietet *VLab* optional den bekannten Floyd-Steinberg Schattier-Algorithmus an, der durch geschicktes Mischen der wenigen Farben oder Graustufen den Eindruck von viel mehr Farben oder Graustufen vermittelt.

### 5.3.2.8 Schwarzer Hintergrund

**Menü:** Steuern » Umrechnen » Schw. Hintergr.

**Gadget:** Checkbox Hintergrund schwarz im Fenster Umrechnen

**ARexx:** BLACKBACKGROUND ON | OFF

Diese Einstellung bestimmt, ob bei der Umrechnung in ein farbiges Amigabild auf jeden Fall ein schwarzer Hintergrund erzeugt werden soll. Dies ist bei der Erzeugung von Bildern für Animationen oder Diashows sehr praktisch, weil dann die Farbe des Bildschirmrahmens nicht flackert. Sie sollten diese Option deshalb vor allem bei der Umrechnung von Sequenzen einsetzen.

### 5.3.2.9 Anzeige auf Amiga/Retina

**Menü:** Steuern » Umrechnen » Anzeige auf Amiga/Retina

**Gadget:** —

**ARexx:** DISPLAYOUTPUT AMIGA | RETINA

Besitzer der MacroSystem-Grafikkarte *Retina* können an dieser Stelle die Ausgabe der umgerechneten Bilder auf Ihre 24 Bit-Karte umlenken. Dabei kann die *Retina* sämtliche Amiga-Bildschirmmodi (auch HAM, AA-Modi, ...) emulieren.

### 5.3.2.10 Anzahl Farben/Graustufen

**Menü:** —

**Gadget:** Listview im Fenster Umrechnen

**ARexx:** CONVERTDEPTH Farbtiefe

Nachdem der Grundtyp zur Bildumrechnung über das Menü oder die Checkbox-Gadgets im Umrechnen-Fenster bestimmt wurde, werden in dem Listview-Gadget alle in diesem Modus zur Verfügung stehenden Farb- bzw. Helligkeitstiefen angezeigt. Durch Anklicken des gewünschten Eintrags oder durch Senden eines ARexx-Befehls kann der gewünschte Modus gewählt werden. Danach kann die Konvertierung gestartet werden.

### 5.3.2.11 Farbtreue

**Menü:** —

**Gadget:** —

**ARexx:** COLORFIDELITY Zahl

Bei der Konvertierung eines YUV-Bildes in ein HAM-Bild kann der benutzte Algorithmus dahingehend beeinflusst werden, ob bei der Umrechnung mehr Wert auf exakte Helligkeits- oder präzisere Farbwiedergabe gelegt wird. Dieser Parameter kann nur über einen ARexx-Befehl verändert werden. Ist der Wert von „Zahl“ 0, wird auf bessere Helligkeit geachtet. 254 muß für eine optimale Farbtreue eingestellt werden. Zahlen dazwischen verlagern die Gewichtung.

### 5.3.3 Sequenz

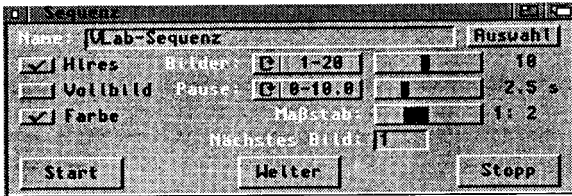


Abbildung 5.7: Das Sequenz-Fenster

#### 5.3.3.1 Auswahl

**Menü:** Steuern >> Sequenz >> Namen wählen...

**Gadget:** Button Auswahl im Fenster Sequenz

**ARexx:** SEQUENCENAME

Bevor eine Sequenz digitalisiert und abgespeichert werden kann, muß ein Name dafür gewählt werden. Es wird ein File-Requester geöffnet, über den der Pfad und der Name für die Sequenz gewählt werden kann. Der Name bildet den Grundnamen für alle Bilder, an diesen wird eine fortlaufende Nummer angehängt.

#### 5.3.3.2 Name

**Menü:** —

**Gadget:** Text Name im Fenster Sequenz

**ARexx:** —

Hierbei handelt es sich um ein Text-Gadget, in dem der für die aktuelle Sequenz benutzte Grundname angezeigt wird. Dieser wurde zuvor mit „Auswahl“ ausgesucht.

### 5.3.3.3 Anzahl der Bilder

**Menü:** —

**Gadget:** Slider Bilder im Fenster Sequenz

**ARexx:** SEQUENCELENGTH Anzahl

Eine Sequenz kann aus 1 bis 9999 Bildern bestehen. Die Anzahl wird über das Slider-Gadget eingestellt. Die aktuelle Anzahl wird gleichzeitig direkt angezeigt. Wie an anderer Stelle bereits beschrieben, kann durch Klicken neben den Mover (Schieberegler) die Zahl der Bilder in Einzelschritten erhöht oder erniedrigt werden. Mit dem Cycle-Gadget links des Slider-Gadgets kann der Bereich, in dem der Schieberegler arbeitet, in Stufen zwischen 1–10 und 1–9999 eingestellt werden. Steht die Bildanzahl auf eins, kann über die Weiter-Operation eine Einzelbildaufnahme mit automatischer Speicherung erfolgen.

### 5.3.3.4 Pause

**Menü:** —

**Gadget:** Slider Pause im Fenster Sequenz

**ARexx:** SEQUENCEDELAY SECS | MINS | TICKS Zeit

Hiermit wird der Zeitabstand eingestellt, in dem die Bilder bei der Sequenzaufnahme hintereinanderfolgen. Die eingestellte Zeitspanne kann natürlich nur dann eingehalten werden, wenn sie länger als die zur Speicherung benötigte Zeit ist. Die Zeit, die zur Speicherung der Bilder benötigt wird, ist also in dieser Pausenzeit enthalten.

Mit dem Cycle-Gadget vor dem Regler kann der Zeitbereich für den Slider rechts in sechs Stufen zwischen 0–10.0 s und 0–9999 s eingestellt werden. Bei der kleinsten Einstellung kann die Pausenzeit auf  $\frac{1}{10}$  Sekunden, in den anderen Bereichen auf eine Sekunde genau eingestellt werden. Eine exaktere Einstellung der Pausenzeit kann über den ARexx-Befehl erfolgen.

Bei Pausenzeiten von mehr als zwei Sekunden wird zwischenzeitlich die Hardware wieder freigegeben, so daß in der Zeit zwischen der Aufzeichnung zweier Bilder die Hardware anderweitig (z.B. von der Monitorfunktion) benutzt werden kann.

### 5.3.3.5 Nächstes Bild

**Menü:** —

**Gadget:** Number Nächstes Bild im Fenster Sequenz

**ARexx:** —

An dieser Stelle wird die Nummer des Bildes angezeigt, das als nächstes digitalisiert wird. Man kann hier gut verfolgen, wie weit die Aufzeichnung einer Sequenz schon gediehen ist, und wie schnell die Aufzeichnung der Sequenz bei den gewählten Einstellungen erfolgt. Dieses Gadget dient z.Zt. nur der Ausgabe.

### 5.3.3.6 Maßstab

**Menü:** —

**Gadget:** Slider Maßstab im Fenster Sequenz

**ARexx:** SEQUENCESCALE Verkleinerungsfaktor

Der Maßstab bestimmt die Größe der abgespeicherten Sequenz-Bilder und kann in Stufen zwischen 1:1 und 1:4 eingestellt werden. Bei 1:1 werden die Bilder in der vollen Größe abgespeichert. Bei 1:4 besitzen die Bilder nur ein Viertel der Höhe und der Breite des normalen Bildes. Der Maßstab wirkt sich deutlich auf die Geschwindigkeit der Sequenz-Funktion und den Speicherplatzbedarf der Bilder aus. Während ein Interlace-Bild im Maßstab 1:1 ca. 600 kB Speicher benötigt, belegt ein Bild im Maßstab 1:3 nur ca. 67 kB. Bevor man die Skalierung einschaltet, sollte zuerst das Vollbild ausgeschaltet werden, weil damit auch schon die vertikale Auflösung und der Speicherplatzbedarf halbiert wird, und so auch Probleme mit bestimmten Bildquellen oder schnell bewegten Objekten vermieden werden. Die Sequenz-Funktion arbeitet besonders schnell im Nicht-Interlace-Betrieb und bei starker Verkleinerung.

### 5.3.3.7 Start

**Menü:** Steuern » Sequenz » Start



**Gadget:** Button Start im Fenster Sequenz

**ARexx:** SEQUENCE START

Das Digitalisieren, Einlesen und Abspeichern der einzelnen Bilder einer Sequenz wird gestartet. Dabei versucht *VLab* eine möglichst hohe Wiederholgeschwindigkeit zu erreichen. Der wichtigste Faktor für die Geschwindigkeit dieser Funktion ist dabei der Zeitbedarf für das Sichern der Bilder auf einem Speichermedium. Auf einer langsamen Festplatte läßt sich nur ca. ein Bild pro Sekunde abspeichern, bei schnellen Festplatten oder der RAM-Disk kann man Raten von über 5 Bildern pro Sekunde erreichen.

Es ist zu beachten, daß die erzeugten Dateien nicht das sonst von *VLab* benutzte YUVN-Format verwenden, sondern aus Geschwindigkeitsgründen in einem internen, undokumentierten *VLab*-Format gespeichert werden. Dieses Format kann nur von *VLab* selbst gelesen werden, die Dateien können aber durch Laden und erneutes Abspeichern leicht in das YUVN-Format umgewandelt werden. Dies geschieht am einfachsten mit einem ARexx-Script. Ein entsprechendes ARexx-Script-File („VLABtoYUVN.rx“) ist in der Makros-Schublade von *VLab* zu finden.



### 5.3.3.8 Weiter

**Menü:** Steuern » Sequenz » Weiter

**Gadget:** Button Weiter im Fenster Sequenz

**ARexx:** SEQUENCE CONTINUE

Die Aufzeichnung einer unterbrochenen Sequenz kann hiermit jederzeit fortgesetzt werden. Die Bildfolge setzt bei der im Gadget nächstes Bild angezeigten Bildnummer wieder ein. Ist die vorherige Sequenz bis zum vorgesehenen Ende abgelaufen, wird erneut die volle Anzahl an Bildern digitalisiert, jedoch ab der aktuellen Bildnummer.

### 5.3.3.9 Stopp

**Menü:** Steuern » Sequenz » Stopp

**Gadget:** Button Stopp im Fenster Sequenz

**ARexx:** SEQUENCE STOP

Die Aufzeichnung einer laufenden Sequenz kann hiermit jederzeit unterbrochen werden.

### 5.3.3.10 HiRes

**Menü:** Steuern » Sequenz » HiRes

**Gadget:** Checkbox Hires im Fenster Sequenz

**ARexx:** SEQUENCEMODE HIRES | LORES

Hiermit wird die horizontale Auflösung (HiRes, LoRes) der aufzuzeichnenden Sequenz eingestellt. Eine Bildfolge in LoRes benötigt dabei nur halbsoviel Speicherplatz auf dem Speichermedium und kann auf Grund der geringeren Datenmenge schneller abgespeichert werden.

### 5.3.3.11 Vollbild

**Menü:** Steuern » Sequenz » Vollbild

**Gadget:** Checkbox Vollbild im Fenster Sequenz

**ARexx:** SEQUENCEMODE LACE | NONLACE

Ist Vollbild ausgewählt, wird die Sequenz in Interlace (also mit Vollbildern statt Halbbildern) aufgenommen. Das Vollbild besitzt die doppelte vertikale Auflösung des Halbbildes, braucht dafür aber auch den doppelten Speicherplatz. Die erreichbare Bildrate beim Non-Interlace Betrieb ist wesentlich höher als beim Interlace-Betrieb.

### 5.3.3.12 Farbe

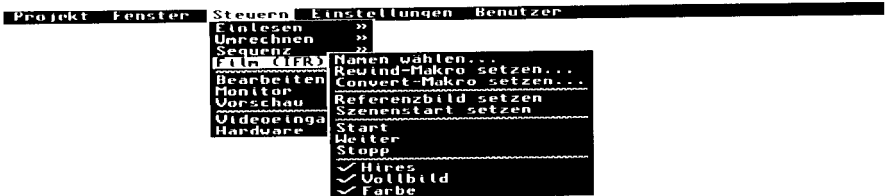
Menü: **Steuern** » **Sequenz** » **Farbe**

Gadget: **Checkbox** Farbe im Fenster Sequenz

ARexx: **SEQUENCEMODE BW | COLOR**

Nur wenn der Menüpunkt Farbe angewählt ist, wird die Sequenz in Farbe aufgezeichnet. Ist er nicht aktiviert, wird nur in schwarzweiß digitalisiert. Das spart sowohl etwas Zeit, als auch Speicherplatz.

### 5.3.4 Film (IFR)



Menü: **Fenster** » **Film (IFR)**...

4

Gadget: —

ARexx: **OPENWINDOW IFR**

Durch dieses Untermenü können die Kommandos des IFR über Menüpunkte gesteuert werden. Mit Hilfe der „Interleaved Frame Recording“ genannten Methode können Echtzeitsequenzen durch mehrfache, verschachtelte Aufnahmevorgänge erzeugt werden. Die Bedienung dieser speziellen Form der Sequenzaufnahme wird im Abschnitt 6.6 ab Seite 87 beschrieben.

## 5.3.5 Bearbeiten



### 5.3.5.1 DeInterlace

Menü: **Steuern** » **Bearbeiten** » **DeInterlace**



Gadget: —

ARexx: **DEINTERLACE 10**

Mit der DeInterlace-Funktion können die „verwishten“ Bereiche in einem Bild beseitigt werden, die auftreten, wenn ein Vollbild aufgenommen wurde und sich ein Bildobjekt zwischen den beiden Halbbildern bewegt hat.

Wenn in der Fernsehtechnik wirklich nur mit Vollbildern gearbeitet würde, dürfte dieser Effekt eigentlich gar nicht auftreten — also wenn wirklich alle 1/25s ein Vollbild aufgenommen würde und dann die beiden Halbbilder einzeln hintereinander übertragen werden würden. Tatsächlich wird jedoch bei den meisten Sendungen alle 1/50s ein unterschiedliches Halbbild aufgenommen, was nun zur Folge hat, daß in einem digitalisierten Vollbild bewegte Objekte verzerrt aussehen. Nur wenige Filme werden wirklich mit 25 Bildern pro Sekunde gesendet; man kann diese Sendungen leicht erkennen, da dort bei einem digitalisierten Vollbild auch bewegte Objekte scharf erscheinen.

Das eben geschilderte Problem kann mit Hilfe der DeInterlace-Funktion beseitigt werden. Nach Untersuchung des Vollbildes wird an allen Stellen, an denen sich ein Objekt bewegt hat, versucht, die Bildinformation des zweiten Halbbildes an diesem Ort aus dem ersten Halbbild herzuleiten. Dabei sinkt in diesem Bereich die vertikale Auflösung bewegter Objekte auf die Auflösung eines Halbbildes. Bereiche, in denen keine Bewegung stattfand, wo also kein Versatz zwischen den einzelnen Zeilen aufgetreten ist, werden nicht verändert, dort besitzt das digitalisierte Bild weiterhin die volle vertikale Auflösung.

Weitere Informationen zur DeInterlace-Funktion finden Sie in Abschnitt 6.2.

### 5.3.5.2 HiRes nach LoRes

**Menü:** Steuern » Bearbeiten » HiRes nach LoRes

**Gadget:** —

**ARexx:** YUVTOLORS

Mit diesem Kommando läßt sich die horizontale Auflösung eines im YUV-Format vorliegenden Videobildes halbieren. Durch Einsatz einer Rechenoperation ist die erzielte Bildqualität höher, als wenn nur jeder zweite Punkt weggelassen worden wäre.

### 5.3.5.3 YUV-Bild sichern

**Menü:** Steuern » Bearbeiten » YUV-Bild sichern

C

**Gadget:** —

**ARexx:** COPYTOUND0

Das digitalisierte YUV-Bild wird zusätzlich in einen Zwischenpuffer kopiert, aus dem es später wieder zurückgeholt werden kann. Achtung: Diese Funktion benötigt zusätzlichen Speicher entsprechend der Größe des digitalisierten Bildes!

### 5.3.5.4 Bild zurückholen

**Menü:** Steuern » Bearbeiten » Bild zurückholen

U

**Gadget:** —

**ARexx:** UNDO

Das Bild im normalen YUV-Bildspeicher wird durch das Bild aus dem Undo-Puffer ersetzt. Daraufhin wird der Puffer gelöscht und der belegte Speicher wieder freigegeben.

### 5.3.5.5 Bild automatisch sichern

**Menü:** Steuern » Bearbeiten » Bild autom. sichern

**Gadget:** —

**ARexx:** AUTOUND0 ON | OFF

Wurde diese Option aktiviert, wird beim Einlesen eines neuen Bildes das zuvor digitalisierte Videobild in den Undo-Puffer geschrieben und kann von dort bei Bedarf wieder zurückgeholt werden. Da jedes Bild bis zu 600 kB Speicher benötigt, empfiehlt es sich, diese Funktion zu deaktivieren, wenn der Arbeitsspeicher knapp wird.

## 5.3.6 Monitor



Mit Hilfe des Monitors kann der Anwender das empfangene Videoprogramm fortwährend auf dem Bildschirm des Computers verfolgen. Normalerweise ist der Monitor ein kleines Fenster auf dem *VLab*-Screen, in dem das laufende Videobild mit der Farbtiefe des aktuellen Screens wiedergegeben wird.

Wenn *VLab* auf einem eigenen Screen arbeitet, erfolgt die Anzeige mit den zur aktuellen Screentiefe gehörenden Graustufen. Läuft *VLab* dagegen auf dem Workbench-Screen, kann die Darstellung nur in den zur Verfügung stehenden Farben der Workbench geschehen. Es werden allerdings zur Umstellung der WB-Farben vorgefertigte Farbpaletten mit Graustufen mitgeliefert.

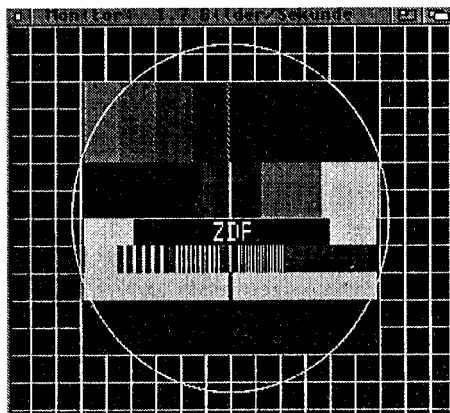


Abbildung 5.8: Das **Monitor**-Fenster

Der Inhalt des Monitorbildes wird je nach Bildgröße, Graustufen und Rechenleistung des eingesetzten Amiga ca. 1–15 mal in der Sekunde aktualisiert. Auf vielfachen Kundenwunsch wird ab *VLab* 2.0 die aktuelle Bildwiederholrate in der Titelleiste des Monitorfensters ausgegeben.

Bei zeitaufwendigen Operationen innerhalb von *VLab*, wie z.B. dem Umrechnen oder dem Speichern von Bildern, wird die Monitorausgabe normalerweise angehalten, um diese Funktionen nicht unnötig zu verlangsamen. Weitere Informationen zum Monitorbetrieb finden Sie in Abschnitt 6.1 sowie beim ARexx-Befehl BLITMODE.

### 5.3.6.1 Betrieb

**Menü:** Steuern » Monitor » Betrieb

M

**Gadget:** —

**ARexx:** MONITOR ON | OFF

Die Monitorfunktion wird hiermit ein- oder ausgeschaltet.

### 5.3.6.2 Monitor-Betriebsart

**Menü:** Steuern » Monitor » Betriebsart

**Gadget:** —

**ARexx:** MONITOROUTPUT SMALLWINDOW | LARGEWINDOW  
...RETINALORES | RETINAHIRES | RETINACOLOR

Mit diesem Teil des Monitor-Untermenüs bestimmt der Benutzer, auf welche Weise der Monitorbetrieb erfolgen soll. In der Regel kann er nur die Größe des Monitorfensters auf „klein“ oder „groß“ einstellen. Ist das Monitorbild auf „groß“ geschaltet, ist die Bildwiederholrate natürlich geringer, es werden jedoch mehr Einzelheiten erkennbar.

Ist der Anwender jedoch in Besitz der MacroSystem-Grafikkarte *Retina*, kann er die Monitorausgabe auch auf diese Grafikkarte umlenken. Dabei stehen ihm drei verschiedene Darstellungsmodi zur Verfügung: Schwarzweiß mit niedriger Auflösung, schwarzweiß in hoher Auflösung und farbig. Die Software setzt einen zweiten Bildpuffer ein, um Bewegung möglichst ruckelfrei erscheinen zu lassen (Double-Buffering).

Wie immer gilt: Je besser die Ausgabequalität ist, umso geringer ist Bildwiederholfrequenz. **Warnung:** Bei der Zorro II-Version von *VLab* funktioniert die schnellste Ausgabemethode nicht in jeder Systemkonfiguration (A 3000/4000, umgerüstete A 2000).

### 5.3.6.3 Hohe Priorität

Menü: **Steuern** » **Monitor** » **Hohe Priorität**

Gadget: —

ARexx: **MONITORPRI HIGH | LOW**

Diese Option erlaubt es, die Priorität des Monitorprozesses zu erhöhen. Der Monitorprozeß läuft gewöhnlich mit einer Priorität, die um fünf niedriger ist als die Priorität von *VLab* (normalerweise 0). Wird die Option *Hohe Priorität* gewählt, erhält der Monitorprozeß die gleiche Priorität wie *VLab*. Das hat zur Folge, daß das Monitorbild beim Umrechnen oder Speichern von Bildern nicht mehr angehalten wird sondern weiterläuft. Dies ist eigentlich nur sinnvoll, wenn das Monitorfenster als Ersatz für einen „richtigen“ Bildschirm dienen muß.

Die Ausführungszeit von rechenintensiven *VLab*-Funktionen sowie auf dem Amiga parallel arbeitenden Prozessen wird durch eine hohe Monitorpriorität demnach deutlich verlangsamt.

### 5.3.7 Vorschau



Sobald im Speicher ein YUV-Bild vorhanden ist, kann dieses relativ schnell in einem Fenster, im Hintergrund oder auf einem eigenen Screen angezeigt werden. Diese Funktion soll die Kontrolle darüber ermöglichen, ob das gewollte Bild eingelesen wurde, ob die Bildqualität stimmt, ob ein „Deinterlace“ durchgeführt werden muß oder ähnliches. Die Darstellung geschieht in der für den jeweiligen Screen gewählten Auflösung, es werden jedoch aus Zeitgründen keine besonderen Anstrengungen unternommen, die Ausgabequalität zu optimieren.

### 5.3.7.1 Vorschau anzeigen

Menü: **Steuern** » **Vorschau** » **Vorschau anzeigen**

P

Gadget: **Button Vorschau im Fenster Einlesen**

ARexx: **PREVIEW**

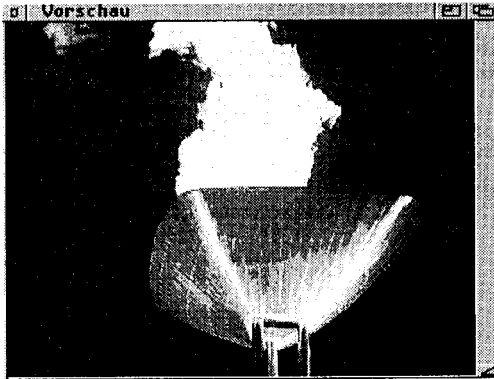


Abbildung 5.9: Ein typisches **Vorschau**-Fenster

Von dem momentan im Speicher befindlichen YUV-Bild wird eine Vorschau berechnet und angezeigt. In welcher Form das Vorschau-Bild angezeigt werden soll, ist vorher über das Menü **Steuern** » **Vorschau** oder über ARexx-Kommandos festzulegen. Es gilt zu beachten, daß das angezeigte Bild nicht die maximal möglich Qualität zeigt, weil hier bei der Umrechnung besonderen Wert auf eine hohe Geschwindigkeit gelegt wurde. Um eine qualitativ hochwertige Umrechnung der YUV-Bilder ins Amiga-IFF-Format zu erhalten, müssen Funktionen aus dem Fenster **Umrechnen** benutzt werden, die allerdings länger dauern. Vorschau-Bilder können weder abgespeichert noch anderweitig weiterbearbeitet werden.

### 5.3.7.2 Vorschau löschen

Menü: **Steuern** » **Vorschau** » **Vorschau löschen**

Gadget: —

ARexx: **ERASEPREVIEW**

Man kann mit diesem Menüpunkt die angezeigte Vorschau explizit löschen. Insbesondere bei Betrieb im Bildschirm-Hintergrund kann das Bild eventuell stören. Bei der Vorschau auf der *Retina*-Grafikkarte wird der benutzte Bildspeicher wieder freigegeben (siehe Abschnitt 5.3.7.6).



### 5.3.7.3 Vorschau automatisch starten

**Menü:** Steuern » Vorschau » Vorschau automatisch  
**Gadget:** —  
**ARexx:** AUTOPREVIEW ON | OFF

Der Menüpunkt Vorschau Automatisch bewirkt die Anzeige der Vorschau sofort nach dem Einlesen eines Videobildes oder nach der Bearbeitung des Bildes mit der DeInterlace-Funktion.

### 5.3.7.4 Vorschau Modus

**Menü:** Steuern » Vorschau » Skaliert, Vollbild, Farbe  
**Gadget:** —  
**ARexx:** PREVIEWMODE SCALE | NOSCALE  
PREVIEWMODE LACE | NONLACE  
PREVIEWMODE COLOR | BW

Neben der Wahl des Ausgabemediums für die Vorschau-Bilder kann auch die Art der Anzeige bestimmt werden. Skaliert bedeutet, daß das Bild auf die Größe des Ausgabefensters verkleinert wird. Die Skalierungsfunktion vergrößert das Bild nicht. Es kann vorkommen, daß das anzuzeigende Bild kleiner ist als das gewählte Fenster oder der Hintergrund. Dieser wird dann nicht ganz ausgefüllt. Die Funktion arbeitet natürlich nur bei der Ausgabe in einem eigenen Fenster oder auf dem Hintergrund, wobei man sich entscheiden muß, ob man nur einen Bildausschnitt sehen möchte oder das ganze Bild, dann aber verkleinert. Die Verkleinerung benötigt allerdings einiges an Rechenzeit. Ist Vollbild gewählt, wird das Bild, falls es ein Vollbild ist, als Vollbild angezeigt, andernfalls nur als Halbbild, was etwas schneller geht und auch Speicher spart. Erfolgt die Ausgabe auf dem VLab-Screen, also in einem Fenster oder auf dem Hintergrund, und besitzt dieser Screen mehr als 300 Zeilen wie z.B. Interlace oder Productivity, so werden, wenn Vollbild nicht ausgewählt ist oder ein Halbbild digitalisiert wurde, nur maximal die einem Halbbild entsprechende Anzahl an Zeilen dargestellt. Es kann also vorkommen, daß das Vorschau-Bild nicht die gesamte Höhe des Vorschau-Fensters oder des Hintergrundes einnimmt.

Weiterhin kann noch Farbe gewählt werden, die Umrechnung dauert dann allerdings deutlich länger, da das Bild bei der Wiedergabe auf einem eigenen Screen in HAM oder bei der Ausgabe auf eine Grafikkarte in 24 Bit RGB konvertiert werden muß. Erfolgt die Darstellung auf dem VLab-Screen, ist diese Auswahl ohne Wirkung; die Ausgabe erfolgt dann immer in Graustufen.

### 5.3.7.5 Vorschau Anzeigeform

**Menü:** Steuern » Vorschau » Anzeigeform

**Gadget:** —

**ARexx:** PREVIEWOUTPUT WINDOW | BACKGROUND | SCREEN  
PREVIEWOUTPUT HARLEQUIN | RETINA

Die Ausgabe des Vorschau-Bildes kann auf verschiedene Arten erfolgen. Ist Fenster gewählt, wird das gerade eingelesene Bild in einem eigenen Fenster auf dem aktuellen Screen ausgegeben. Ist dieses Fenster noch nicht offen, wird es zuvor geöffnet. Da das Fenster die Farben des jeweiligen Screens verwenden muß, erhält man normalerweise eine Schwarzweiß-Ausgabe. Die Anzahl der Graustufen kann zwei, vier, acht oder 16 betragen. Bei acht Graustufen ist bereits eine gut erkennbare Darstellung des digitalisierten Bildes zu erhalten. Läuft *VLab* auf dem Screen der Workbench, ist (je nach Farbpalette) nur eine Falschfarbendarstellung möglich. Geeignete Graustufen-Paletten sind jedoch im Unterverzeichnis *Presets/VLab* in der *Prefs*-Schublade zu finden.

Die Größe des Fenster kann beliebig gewählt werden. Es wird immer der in der Mitte des Gesamtbildes gelegene Ausschnitt gezeigt, es sei denn, es wurde auf Skalierte Vorschau geschaltet. In diesem Fall wird das Bild auf die Fenstergröße verkleinert. Die Skalierung benötigt allerdings zusätzliche Rechenzeit.

Die zweite Möglichkeit besteht in der Ausgabe des Bildes auf dem Hintergrund des *VLab*-Screens. Dort gilt in Bezug auf Graustufen und Skalierung ebenfalls das oben Gesagte, obwohl eine Skalierung hier oft nicht nötig ist. In diesem Modus nimmt das angezeigte Bild den gesamten Hintergrund ein und wird nur stellenweise von geöffneten *VLab*-Fenstern verdeckt.

Bei einer Ausgabe auf einem Screen wird das Bild unabhängig vom Typ des *VLab*-Screens auf einem eigenen Bildschirm angezeigt und zwar je nach Modus in Schwarzweiß mit 16 Graustufen oder als farbiges HAM-Bild. Hier noch einmal der Hinweis, daß die nur mäßige Qualität des HAM-Vorschau-Bildes auf Grund der Geschwindigkeit der Umrechnung an dieser Stelle hingenommen werden muß. Eine wirklich originalgetreue Wiedergabe ist nur über die Funktionen *Umrechnen* und *Anzeige* zu erreichen.

Die letzte Methode der Darstellung ist die Ausgabe auf einer 24 Bit-Grafikkarte. Dazu kann sowohl die „Harlequin“ von ASC (Amiga-Center-Scotland), als auch die *Retina* von MacroSystem eingesetzt werden. Mit dieser Hardware ist die Darstellung eines RGB-Bildes mit bis zu 16 Millionen Farben in der vollen Auflösung des Fernsehbildes möglich. Da ein solches RGB-Bild sogar etwas mehr Farbtöne enthalten kann, als im YUV-Videosignal vorhanden sind, wird mit diesen Karten das digitalisierte Bild in seinem ganzen Farbenreichtum und seiner vollen Auflösung wiedergegeben.

Weil die „Vorschau“ auf solchen Grafikkarten bereits den gesamten Informationsgehalt des eingelesenen Bildes darstellen kann, entfällt natürlich ein weiteres, zeitaufwendiges Umrechnen des Bildes.

### 5.3.7.6 Hinweise zur Retina-Vorschau

Die Treibersoftware der *Retina*-Grafikkarte besitzt eine eigene Screenverwaltung, die ähnlich wie die Verwaltung der Amiga-Screens funktioniert. Das bedeutet, daß der vorhandene Grafikspeicher mit mehreren Bildern belegt werden kann, solange genügend RAM bereitsteht. Programme, die Ausgaben auf der *Retina* machen wollen, müssen daher für ihre Screens Bildspeicher von der *Retina*-Library anfordern.

Dabei liegen die Grafiken „übereinander“, denn es kann natürlich nur eines dieser Bilder auch angezeigt werden. Welcher der im Bildspeicher vorliegenden Screens gerade „vorn“ ist und deshalb dargestellt wird, kann mit einer Tastenkombination über ein mitgeliefertes „Commodity“ zyklisch umgestellt werden.

Auch die *Retina*-Unterstützung der *VLab*-Software (Monitor, Vorschau) fügt sich in dieses Konzept ein. Wenn also mit der Vorschau-Funktion ein Videobild auf der Grafikkarte ausgegeben wird, muß für den 24 Bit-Screen Bildspeicher belegt werden. Dieses RAM steht für andere Anwendungen (z.B. für den Monitorbetrieb) erst wieder zur Verfügung, wenn der Screen geschlossen und damit der Speicher wieder freigegeben wird.

Die *VLab*-Vorschau befindet sich solange im Grafikspeicher der *Retina*, bis der Menüpunkt Vorschau löschen bzw. das AREXX-Kommando ERASEPREVIEW aufgerufen wird. Erst dann wird das digitalisierte Bild gelöscht und das belegte Video-RAM kann anderweitig verwendet werden.

Die Vorschau auf der *Retina* kann übrigens auch mit der rechten Maustaste beendet werden.

### 5.3.8 Videoeingang

**Menü:** Steuern » Videoeingang » 1, 2, 3, 4, 5, 6  
**Gadget:** MX Eingang im Fenster Quelle definieren  
**ARExx:** INPUT 1..6



Jede VLab-Hardware besitzt mehrere (2 oder 3) Videoeingangsbuchsen. Über diesen Menüpunkt oder das MX-Gadget Eingang im Fenster Quelle definieren kann einer davon aktiviert werden. Ist ein zweiter VLab-Digitizer im Rechnersystem vorhanden, verwenden dessen Buchsen die nächsten drei Nummern. Die softwareseitige Zuordnung der Videoanschlüsse kann man auf Seite 16 nachlesen.

### 5.3.9 Hardware



#### 5.3.9.1 Videorecorder

**Menü:** Steuern >> Hardware >> Videorecorder

**Gadget:** Checkbox VTR im Fenster Quelle definieren

**ARexx:** VTRMODE ON | OFF

Dieser Menüpunkt entspricht der Funktion der Checkbox VTR im Fenster Quelle definieren und muß dann eingeschaltet werden, wenn von einem Videorecorder digitalisiert werden soll. Näheres zu dieser Funktion findet man in Abschnitt 5.2.7.4.

### 5.3.9.2 VLab-Reset

**Menü:** Steuern » Hardware » VLab-Reset

**Gadget:** —

**ARexx:** VLABRESET

Unter Umständen kann es vorkommen, daß die Videochips von *VLab* das empfangene Bild nicht mehr stabil darstellen können. Dieser Fall könnte insbesondere dann auftreten, wenn „VSC“ eingeschaltet ist. Deshalb kann der Hardware ein Resetsignal gesendet werden, um das Bild wieder einrasten zu lassen.

### 5.3.9.3 VLab/par einbinden

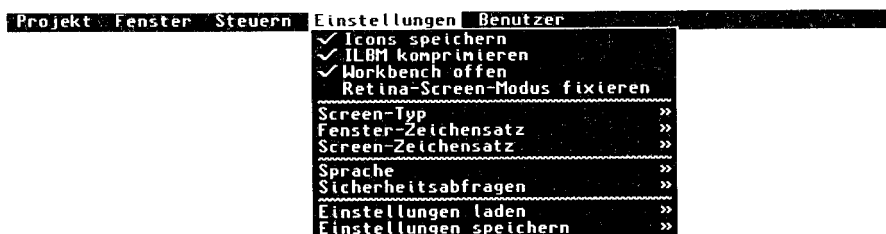
**Menü:** Steuern » Hardware » VLab/par einbinden

**Gadget:** —

**ARexx:** BINDVLABPAR

Sollte das Programm beim Start das Vorhandensein von *VLab/par* nicht erkannt haben, weil etwa die Hardware noch ausgeschaltet war, kann der Anwender mit diesem Kommando im nachhinein die Betriebsbereitschaft des externen Gerätes melden. Denken Sie daran, daß *VLab/par* dann die jeweils höchsten benutzten Eingangsbuchsen belegt!

## 5.4 Menü: Einstellungen



### 5.4.1 Icons speichern

Menü: **Einstellungen** >> **Icons speichern**

Gadget: —

ARexx: **SAVEICONS ON | OFF**

Diese Einstellung entscheidet darüber, ob beim Abspeichern von *VLab*-Dateien ein zugehöriges Piktogramm erzeugt wird oder nicht. Die von *VLab* verwendeten Icons sehen wie eine Videokassette aus, auf deren Aufkleber der jeweilige Bildtyp zu erkennen ist.

### 5.4.2 ILBM komprimieren

Menü: **Einstellungen** >> **ILBM komprimieren**

Gadget: —

ARexx: **ILBMCOMPRESSION ON | OFF**

Mit diesem Menüpunkt läßt sich die Kompression von IFF-ILBM-Bilddateien ein- und ausschalten. Einerseits kommen einige Amiga-Programme nur mit komprimierten Dateien zurecht, andererseits kostet die Kompression Rechenzeit und spart kaum Speicherplatz. Manche Bilder werden durch die Komprimierung sogar länger.

### 5.4.3 Workbench öffnen

Menü: **Einstellungen** >> **Workbench öffnen**

Gadget: —

ARexx: **WORKBENCH OPEN | CLOSE**

Wird diese Option deaktiviert, versucht das Programm, den Screen der Workbench zu schließen. Gelingt dies, kann der für diesen Bildschirm benötigte Speicher freigegeben werden. Der Workbench-Screen kann nur geschlossen werden, wenn auf ihm keine Fenster anderer Applikationen offen sind. Kann der WB-Screen nicht geschlossen werden, erfolgt eine entsprechende Fehlermeldung.

### 5.4.4 Retina-Screen-Modus fixieren

**Menü:** Einstellungen » Retina-Screen-Modus fixieren

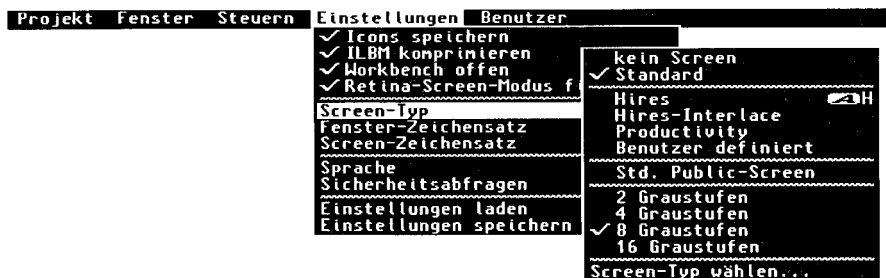
**Gadget:** —

**ARexx:** LOCKRETINASCREENMODE ON | OFF

Arbeitet *VLab* mit der Grafikkarte *Retina* zusammen, können mit diesem Menüeintrag die aktuellen Betriebsarten der *Retina* für die weitere Verwendung durch die *VLab*-Software fixiert werden.

Dadurch wird es möglich, mit Hilfe von „Retina-Screenmode“ für *VLab* geeignete Bildschirmmodi auszuwählen. Solange diese Option anschließend aktiviert ist, werden die gewünschten Bildschirmauflösungen unabhängig von Veränderungen in „Retina-Screenmode“ beibehalten.

### 5.4.5 Screen-Typ



Dieses Untermenü dient der Bestimmung diverser Parameter, mit denen die Art des *VLab*-Bildschirms festgelegt wird.

#### 5.4.5.1 Kein Screen

**Menü:** Einstellungen » Screen-Typ » kein Screen

**Gadget:** —

**ARexx:** SCREENMODE WORKBENCH

Die Einstellung `kein Screen` bewirkt, daß *VLab* auf dem Screen der Workbench arbeitet. Die anderen Menüpunkte bleiben dann ohne Wirkung.

#### 5.4.5.2 Standard

**Menü:** Einstellungen » Screen-Typ » Standard

**Gadget:** —

**ARexx:** SCREENMODE DEFAULT

Wurde Standard gewählt, erhält *VLab* einen eigenen Screen, dessen Eigenschaften denen des Workbench-Screens entsprechen.

#### 5.4.5.3 Videomodus

**Menü:** Einstellungen » Screen-Typ » Videomodus



**Gadget:** —

**ARexx:** SCREENMODE HIRES | HIRESLACE | PRODUCTIVITY

Mit diesen Menüpunkten kann man die Betriebsart der Videochips für den *VLab*-Screen explizit einstellen.

#### 5.4.5.4 Standard Public-Screen

**Menü:** Einstellungen » Screen-Typ » Std. Public-Screen

**Gadget:** —

**ARexx:** DEFAULTPUBSCREEN ON | OFF

Hierdurch läßt sich der *VLab*-Screen zum „Default Public Screen“ erklären, so daß andere Anwendungen relativ leicht Fenster auf ihm öffnen können. Weitere Hinweise zu Public Screens erhalten Sie auf Seite 166.

#### 5.4.5.5 Anzahl Graustufen

**Menü:** Einstellungen » Screen-Typ » X Graustufen

**Gadget:** —

**ARexx:** SCREENDPTH 2 | 4 | 8 | 16

Diese Menüeinträge legen die Anzahl der Farben des *VLab*-Screens fest. Bitte beachten Sie, daß auf einem Productivity-Screen maximal vier Farben dargestellt werden können und daß im Productivity-Betrieb oder bei 16 Farben die Rechenleistung des Amiga durch die hohe DMA-Auslastung vermindert wird.



### 5.4.5.6 Screen-Typ wählen

Menü: Einstellungen » Screen-Typ » Screen-Typ wählen...

Gadget: —

ARexx: SCREENMODE

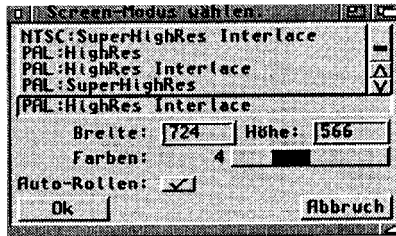
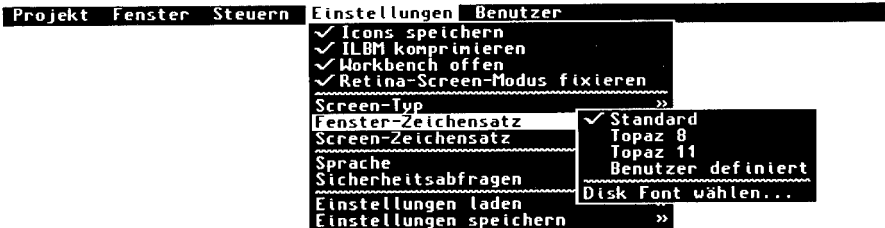


Abbildung 5.10: Der ScreenMode-Requester von OS 2.X

Nach dem Aufruf dieser Funktion öffnet sich ein ScreenMode-Requester. Mit diesem kann man den gewünschten Screen-Modus aus einer Liste herausuchen.

## 5.4.6 Zeichensätze



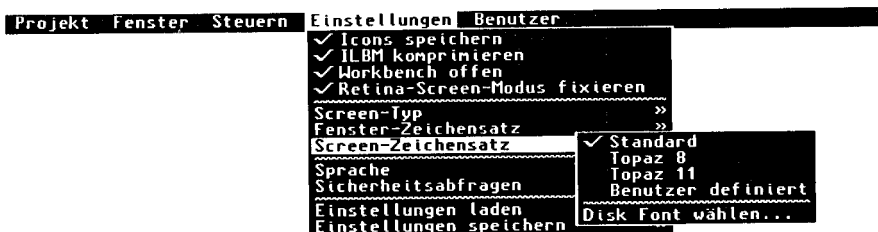
### 5.4.6.1 Fenster-Zeichensatz

Menü: Einstellungen » Fenster-Zeichensatz » Font

Gadget: —

ARexx: FONT

Die Schriftart, die VLab innerhalb seiner Fenster verwendet, kann hier vom Benutzer beliebig eingestellt werden. Als Menüpunkte werden direkt vier Schriften zur Auswahl gestellt.



### 5.4.6.2 Screen-Zeichensatz

**Menü:** Einstellungen » Screen-Zeichensatz » Font

**Gadget:** —

**ARexx:** SCREENFONT

Hiermit wird die Schriftart festgelegt, die *VLab* bei den Titelleisten der Fenster sowie in den Menüs benutzt. Dem Anwender stehen in einem Untermenü vier Schriften direkt zur Verfügung.

### 5.4.6.3 Schriftenmenüs

Standard ist die Schrift, die durch die Einstellung der Workbench-Preferences angegeben wurde. Topaz 8 ist der bekannte Standard-System-Font des Amiga, Topaz 11 die Variante der Schrift, die durch ihre größere Höhe für den Einsatz auf Interlace-Screens vorgesehen ist. Benutzer definiert stellt auf den Font um, der zuvor mit ...-Zeichensatz » Disk Font wählen ausgesucht wurde.

Disk Font wählen startet den Fontauswahl-Requester. Der Start des Requesters kann einen Augenblick dauern, bis die Namen aller Fonts aus dem Fonts:-Verzeichnis eingelesen sind. In dem Font-Requester werden alle Fonts aus dem Fonts:-Verzeichnis im Listview-Gadget links aufgelistet. Wenn Sie dort eine Schrift auswählen, werden in dem kleinen Listview-Gadget rechts alle zur Verfügung stehenden Schriftgrößen angezeigt. Wählen Sie dort eine Größe aus, wird (in dem Feld darunter) ein Probetext dieser Schrift dargestellt.

Der Font-Requester kann auch mit den neuen skalierbaren Fonts der Workbench 2.X umgehen. Hat man sich für einen dieser Fonts entschieden, kann man eine beliebige Fontgröße in dem Feld unterhalb des Listview-Gadgets für die Fontgröße eingeben. Diese Fontgröße wird dann sofort von der Diskfont-Library berechnet und angezeigt, was aber auf einem unbeschleunigten Amiga einige Sekunden dauern kann.

Das Ok-Gadget übernimmt den ausgewählten Font und schließt den Requester. Cancel beendet den Requester ohne eine Auswahl vorzunehmen.

## 5.4.7 Sprache



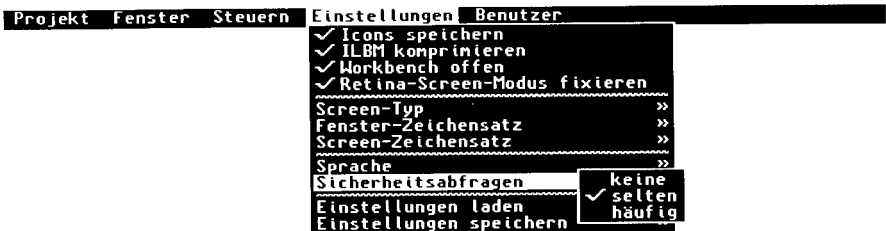
Menü: **Einstellungen** » **Sprache** » **Standard** oder **Sprache**

Gadget: —

ARExx: **LANGUAGE DEFAULT** | **Sprache** | **INTERNAL**

Über dieses Kommando kann ein Wechsel der Sprache vorgenommen werden, die *VLab* bei seinen Programmtexten und der Hilfefunktion benutzt. Bei Verwendung der Option „Standard“ wird die Landessprache eingesetzt, mit der die Workbench gerade arbeitet. Im *VLab*-Programmcode sind englische Texte (ohne Hilfetexte) assembliert; alle anderssprachigen Texte müssen sich im Unterverzeichnis *Catalogs* in jeweils einer eigenen Schublade befinden. Die Sprachumstellung von *VLab* funktioniert auch unter Workbench 2.04, obwohl erst ab Version 2.1 die länderspezifischen Anpassungen (*locale*) unterstützt werden. Muß RAM-Speichersplatz gespart werden, empfiehlt es sich, mit dem AREXX-Befehl **LANGUAGE INTERNAL** auf die eingebauten englischen Texte zurückzugreifen.

## 5.4.8 Sicherheitsabfragen



**Menü:** Einstellungen » Sicherheitsabfragen » Häufigkeit

**Gadget:** —

**ARexx:** SAFETYLEVEL LOW | MEDIUM | HIGH

Hier kann die Häufigkeit der Sicherheitsabfragen eingestellt werden. Bei *keine* treten überhaupt keine Abfragen auf. Bei *selten* erfolgen Sicherheitsabfragen nur an wichtigen Stellen, wie z.B. beim Abspeichern mit der Save-Funktion, bevor eine Datei gleichen Namens überschrieben wird. Bei *häufig* werden Sicherheitsabfragen vor praktisch allen Funktionen gemacht, die bereits bearbeitete Daten zerstören könnten. Dieser Modus ist eigentlich nur für den „blutigen Laien“ gedacht und kann wohl nach meist kurzer Einarbeitungszeit abgeschaltet werden.

### 5.4.9 Einstellungen laden



**Menü:** Einstellungen » Einstellungen laden » Art

**E**

**Gadget:** —

**ARexx:** LOADSETTINGS Name | DEFAULT

Hier können persönliche Voreinstellungen, die vorher abgespeichert wurden, wieder eingeladen werden. Die Einstellungen können von *VLab* aus verschiedenen Quellen bezogen werden.

Voreinstellungen stellt *VLab* auf eine interne Voreinstellung ein, mit der man meist sofort beginnen kann, Bilder aufzunehmen.

Temporäre lädt die Einstellungen aus einem *VLab*-Zwischenspeicher im ENV:-Verzeichnis. Dieses Verzeichnis wird in der Regel vom Betriebssystem auf der Ram-Disk angelegt. Es enthält unter anderem die Konfiguration, die auch beim Aufruf von *VLab* verwendet wurde, solange diese Werte noch nicht mit Einstellungen » Einstellungen speichern » Temporär überschrieben wurden.

Aus Archiv lädt die Einstellungen aus einer *VLab*-Umgebungsvariablen, die sich im ENVARC:-Verzeichnis befindet. Dieses wird vom Betriebssystem so verwaltet, daß die Inhalte auch nach Abschalten des Rechners erhalten bleiben.

Als letzte Möglichkeit können Sie noch unter Benutzer eigene beliebige Einstellungen laden, die vorher unter einem eigenen Namen abgespeichert wurden. Dazu erscheint der File-Requester, der zur Eingabe des Dateinamens auffordert.

### 5.4.10 Einstellungen speichern



Fast alle Einstellungen innerhalb von *VLab*, wie Fensterposition, Dateipfade, geöffnete Fenster etc., lassen sich abspeichern, damit man später wieder schnell auf diese Voreinstellungen zurückgreifen kann.

#### 5.4.10.1 Temporär

**Menü:** Einstellungen » Einstellungen Speichern » Temporär

**Gadget:** —

**ARexx:** SAVESETTINGS ENV:VLab/VLab.Prefs

Die Einstellungen werden in einer Umgebungsvariablen abgelegt und können bis zu einem Reset des Rechners von dort wieder zurückgelesen werden. Bei jedem Programmstart von *VLab* wird dieser Parametersatz verwendet.

#### 5.4.10.2 Archivieren

**Menü:** Einstellungen » Einstellungen Speichern » Archivieren

**Gadget:** —

**ARexx:** SAVESETTINGS ENVARC:VLab/VLab.Prefs

Die Einstellungen werden in einer Environment-Variablen im nicht-flüchtigen ENVARC:-Verzeichnis gesichert. Nach jedem Reset des Rechners werden die Inhalte von ENVARC: und damit auch der *VLab*-Parametersatz nach ENV: umkopiert.

### 5.4.10.3 Speichern als

**Menü:** Einstellungen » Einstellungen Speichern » Speichern als...

**Gadget:** —

**ARexx:** SAVESETTINGS Dateiname

Die Einstellungen werden unter einem beliebigen Namen abgespeichert. Es erscheint ein File-Requester, der nach einem Dateinamen für die Voreinstellungen fragt.

### 5.4.10.4 Automatisch übernehmen

**Menü:** Einstellungen » Einstellungen Speichern » Automatisch übernehmen

**Gadget:** —

**ARexx:** AUTOUSEENV ON | OFF

Diese Option legt fest, ob die temporäre Umgebungsvariable, welche die Einstellungen von *VLab* enthält, automatisch beim Beenden von *VLab* aktualisiert werden soll. Wenn ja, startet *VLab* (bis zum nächsten Reset) immer in der Konfiguration, die beim letzten Verlassen gültig war.

### 5.4.10.5 Automatisch archivieren

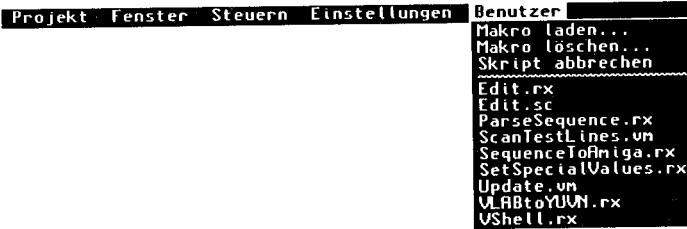
**Menü:** Einstellungen » Einstellungen Speichern » Automatisch archivieren

**Gadget:** —

**ARexx:** AUTOSAVEENV ON | OFF

Das Checkmark entscheidet, ob beim Verlassen von *VLab* die aktuellen Einstellungen im *ENVARC:-*Verzeichnis hinterlegt werden, so daß sie auch nach einem Reset des Rechners zur Verfügung stehen und wieder automatisch geladen werden können. So kann man etwa am nächsten Tag sofort mit der Konfiguration weiterarbeiten, mit der man am Vortag aufgehört hat.

## 5.5 Menü: Benutzer



In diesem Menü können Sie über die Funktion Benutzer » Makro laden eigene Makros in die Liste unterhalb der Menüpunkte eintragen. Diese werden sofort alphabetisch einsortiert und können danach direkt über das Menü gestartet werden. Mehr über Makros und ARexx findet man ab Seite 108.

### 5.5.1 Makro laden

**Menü:** Benutzer » Makro laden...

**Gadget:** —

**ARexx:** LOADMACRO

Mit diesem Kommando wird der Dateiname eines zu ladenden Makros festgelegt. Aus Speicher- und Geschwindigkeitsgründen erfolgt hiermit noch kein Einlesen. Das Makro wird automatisch bei der ersten Benutzung eingeladen. Die in die Makroliste eingetragenen Makronamen werden zusammen mit den sonstigen Einstellungen abgespeichert. VLab-Makros sind Textdateien und müssen mit einem #VLab beginnen. Wird beim Einlesen der Datei kein #VLab gefunden, wird überprüft, ob sie mit /\* ... \*/ beginnt. In diesem Fall liegt ein ARexx-Skript vor, das dann mit Hilfe von SYS:Rexxc/RexxMast abgearbeitet wird. Ansonsten wird versucht, das File als Programm zu starten. Dadurch können über das MACRO-Kommando auch auszuführende Programme (wie etwa ein Editor) oder Batch-Dateien aufgerufen werden.

## 5.5.2 Makro löschen

**Menü:** Benutzer » Makro löschen...

**Gadget:** —

**ARexx:** DELETEMACRO

Die Einträge in der Makroliste können z.Zt. nur durch den *VLab*-Befehl „DELETEMACRO“ entfernt werden. Dieser Menüpunkt öffnet dazu das Fenster *Befehl ausführen*.

## 5.5.3 Skript abbrechen

**Menü:** Benutzer » Skript abbrechen

**Gadget:** —

**ARexx:** BREAK D

Den Abbruch ablaufender Skripts ermöglicht dieser Menüpunkt. Voraussetzung ist jedoch, daß das zu unterbrechende Skript das Abbruchkriterium (Ctrl-D) auch auswertet.



# Kapitel 6

## Praktisches Arbeiten

In diesem Kapitel folgen nähere Erläuterungen zu einigen Funktionen von *VLab*. Dazu kommen einige Tips und Hinweise, die die praktische Arbeit mit diesem Produkt erleichtern sollen.

### 6.1 Der Monitor-Prozeß

Die Priorität des Monitor-Prozesses ist entscheidend für die Arbeitsweise der Monitorfunktion. Normalerweise sollte die Priorität auf „niedrig“ eingestellt sein, damit andere Prozesse, insbesondere *VLab* selbst, bei der Abarbeitung Vorrang haben. Wird z.B. von *VLab* ein Bild berechnet oder eine Datei gespeichert, so wird der Monitorbetrieb gehalten. Ist dagegen Hohe Priorität gewählt, läuft die Monitorfunktion, wenn auch langsamer, weiter. Die Rechenzeit wird dann zwischen *VLab* und dem Monitorprozeß gleichmäßig verteilt, so daß auch das Umrechnen oder Speichern verzögert ausgeführt wird. Es ist etwas Geschmackssache, für welche Methode man sich entscheidet.

Der laufende Monitorprozeß kann unter Umständen auch zu Problemen führen. Arbeitet im Hintergrund ein Programm mit höherer Priorität als der Monitorprozeß, aktualisiert sich die Monitoranzeige solange nicht mehr, bis dieser Prozeß beendet wird (oder in den Wartestatus geht).

Die bei der Monitorausgabe erzielte Bildwiederholrate, aber auch die die erreichbare Bildqualität (Flackern) sowie die dem System verbleibende Rechenkapazität sind von mehreren Parametern abhängig.

- Der Größe des Monitorfensters:  
Bei der Betriebsart „Großes Fenster“ müssen viermal mehr Bildpunkte dargestellt werden.

- Der Anzahl der Farben:  
Sollen 16 Graustufen benutzt werden, muß *VLab* die vierfache Speichermenge übertragen wie bei zwei Graustufen.
- Der Auflösung des Amiga-Screens:  
Bei erhöhter Vertikalauflösung (Interlace) werden doppelt so viele Zeilen verwendet.
- Der Ausgabemethode:  
*VLab* setzt für den Monitorbetrieb verschiedene Ausgabemodi ein, die im Regelfall automatisch ausgewählt werden. Über das AREXX-Kommando „BLITMODE“ kann die Art der Ausgabe auch vom Anwender beeinflusst werden. Man sollte beachten, daß das Monitorfenster nicht von anderen Fenstern überlagert sein sollte (auch nicht teilweise), weil sonst automatisch eine langsame Betriebsart eingestellt wird.
- Der Rechenleistung Ihres Amiga:  
Natürlich bieten Computer mit schnelleren Prozessoren eine größere „Performance“. Die Monitor-Ausgabegeschwindigkeit kann unter Umständen mit der Option „BLITMODE CPU“ noch gesteigert werden.

Programme, die ständig die volle CPU-Aktivität benötigen, sollten nicht gleichzeitig mit *VLab* benutzt werden, da sie auch andere Systemfunktionen beeinträchtigen und das System sehr langsam machen. Leider gibt es aber besonders im Public-Domain-Bereich viele solcher Programme (Virus tester, Malprogramme, Kopierprogramme, ...). Durch Umschalten auf Hohe Priorität kann evtl. Abhilfe geschaffen werden. Besser wäre allerdings der Verzicht auf die gleichzeitige Abarbeitung einer solchen rechenzeitintensiven Anwendung.

## 6.2 Die DeInterlace-Funktion

Bei vielen Signalquellen tritt nach Einlesen eines Vollbildes ein sehr lästiges Flackern im angezeigten Standbild auf. Bei Verwendung eines Display-Enhancers (z.B. der MacroSystem-DeInterlace-Karte) sind in diesem Fall hingegen kammförmige Verzerrungen im dargestellten Bild erkennbar. Die Ursache dafür ist schnell erklärt:

Beim Fernsehen werden immer nur Halbbilder übertragen und zwar 50 Bilder pro Sekunde. Ein komplettes Vollbild besteht somit aus zwei zeitlich nacheinander aufgenommenen Teilbildern. In diesem zeitlichen Versatz liegt nun das Problem. Hat sich in der Zeit zwischen der Aufnahme des ersten und des zweiten Halbbildes ein Objekt oder ein ganzer Bildausschnitt verändert, z.B. bei einer schnellen Bewegung einer Person von

links nach rechts, stimmt die Position in den beiden Bildern nicht mehr überein. Da nach dem Einlesen aber beide Bilder gleichzeitig angezeigt werden, sieht man, um bei dem Beispiel zu bleiben, eine Hälfte der Person gegenüber der anderen Hälfte ganz oder teilweise um einige Millimeter oder sogar Zentimeter versetzt.

Die DeInterlace-Funktion stellt nun fest, welche Bildbereiche betroffen sind und rechnet das Bild an diesen Stellen um. In diesen Bereichen reduziert sich daher die vertikale Auflösung des Bildes. Die hohe Gesamtauflösung des Vollbildes bleibt an allen anderen Stellen jedoch voll erhalten.

Bei Ausführung der DeInterlace-Funktion vom Menü aus, wird ein mittlerer Unterdrückungsgrad benutzt (Wert=10). Andere Werte können beim Aufruf des Befehls über das Eingabefenster oder über AREXX als Parameter angegeben werden. Höhere Werte (Angaben bis 50 sind sinnvoll) bewirken eine Unterdrückung des Flackerns nur in Bereichen mit hohem Kontrast. Kleinere Werte unterdrücken auch das Flimmern in relativ kontrastschwachen Bildbereichen. Aber Vorsicht, bei Werten unter 5 werden auch Bildbereiche verändert, die gar nicht vom Flackern betroffen sind. So niedrige Angaben sollten nur benutzt werden, wenn bei Benutzung höherer Werte sichtbare Schatten im Randbereich der betroffenen Zonen auftreten.

## 6.3 Unterschiedliche Videoquellen

Hier folgt eine Aufzählung verschiedener Signalquellen, die die Einschätzung erleichtern soll, ob man die DeInterlace-Funktion evtl. anwenden muß:

- **Nachrichten, Studioaufnahmen, Shows, TV-Filme, Werbung...**

Diese Programme werden mit Studio-Videokameras im Zeilensprungverfahren (Interlace) auf hochwertigen Bandmaschinen (MAZ) aufgezeichnet. Bei ihnen treten mit hoher Wahrscheinlichkeit Probleme auf, die jedoch von der DeInterlace-Funktion zuverlässig behoben werden können.

- **Kinofilme**

Kinofilme werden meist mit mechanischen Filmkameras produziert, die nur eine andere (geringere) Bildwechselfrequenz erlauben als Videokameras. Deshalb müssen sie vor der Wiedergabe im Fernsehen oder auf Video umkopiert werden. Je nach eingesetztem Verfahren können dabei wieder Interlace-Probleme entstehen.

- **Videokameras, Videorecorder**

Preiswerte Videoausrüstungen können im Standbildbetrieb in der Regel nur Halbbilder wiedergeben, so daß keine Schwierigkeiten zu befürchten sind, aber die Auflösung sehr schlecht ist. Teure Geräte (etwa mit digitalem Standbild) können

allerdings auch Vollbilder produzieren und sind daher auch vom Interlace-Problem betroffen.

Es kann sogar vorkommen, daß man beim Umblenden von einem Bild zu einem anderen das erste Halbbild von aus dem alten und das zweite Halbbild aus dem neuen Bild erhält. In diesem Fall ist jedoch auch die beste Korrektur vergebens.

## 6.4 Festlegung des Bildausschnittes

Die *VLab*-Hardware kann den gesamten Informationsgehalt eines vollen Videobildes digitalisieren und dem Benutzer zur Verfügung stellen. Es können daher im PAL-Betrieb maximal 625 Zeilen mit je 720 Bildpunkten eingelesen werden. Auf einem Fernsehgerät ist jedoch nur ein deutlich kleinerer Bereich für den Betrachter wirklich sichtbar. Bei der Austrahlung von Breitwand-Kinofilmen ist außerdem oben und unten ein (mehr oder weniger breiter) schwarzer Streifen vorhanden. Die *VLab*-Software erlaubt es nun, den gewünschten Bildausschnitt für die einzelnen Signalquellen beliebig festzulegen. Dazu gibt man im Fenster *Quelle* definieren die X- und Y-Koordinaten des linken oberen und des rechten unteren Eckpunktes an (siehe 5.2.7). Durch Ausprobieren der Maximalwerte können Sie den Aufbau Ihres Videobildes leicht studieren. Folgende Effekte werden Ihnen auffallen:

- Schwarze Streifen am linken oder rechten Bildrand  
Diese Bereiche liegen beim Fernseher außerhalb des auf der Bildröhre darstellbaren Ausschnittes und sollten einfach abgeschnitten werden.
- Mit *VLab* ist ein größeren Bildausschnitt nutzbar als auf einem Fernsehgerät.  
Richtig, man kann das Videobild mit „Overscan“ digitalisieren, weil viele Programme noch verwertbare Bildinformationen in den Randbereichen enthalten, die der Fernsehapparat jedoch nicht mehr darstellen kann.
- Schwarze Zeilen am oberen und unteren Bildrand  
Auch dieser Teil des Bildes (vertikale Austastlücke) wird von der Bildröhre nicht genutzt und kann abgeschnitten werden.
- Merkwürdige Signale im obersten Bildbereich  
In diesem Abschnitt des Videobildes befinden sich digitale Zusatzinformationen wie etwa Videotext und VPS. Außerdem sind einige Testsignale des Senders (wie z.B. eine Grautreppe) zu erkennen. Mit Hilfe des *ARexx*-Befehls *VZOOM* oder des Skripts „*ScanTestLines*“ können diese Zeilen einfacher analysiert werden.

Bei der Auswahl eines Ausschnittes können (im möglichen Wertebereich) beliebige Breiten gewählt werden. Intern werden die angegebenen Grenzen gerundet, so daß die Bildbreite um einige Pixel von der gewünschten Breite abweichen kann (siehe 6.8.2). Bei der Angabe der Zeilen wird generell von einem Vollbild ausgegangen. Die Zeilenangaben werden von Programm auf gerade Werte gerundet, so daß beim Einlesen von Vollbildern beide Halbbilder die gleiche Zeilenzahl haben. Andernfalls würden bei der späteren Darstellung auf dem Amiga Probleme auftauchen.

Ein typischer Bildbereich für gewöhnliche Fernsehprogramme ist z.B. der Ausschnitt von (24,32) bis (695,591). Man erhält dann ein Bild aus 560 Zeilen mit je 672 Bildpunkten. Das entspricht in etwa der sichtbaren Fläche einer normalen Fernsehöhre. Eventuell nutzbare Randbereiche (Overscan) werden abgeschnitten — dafür können jedoch keine schwarzen Ränder auftreten. Höhe und Breite sind durch 16 teilbar, damit andere Programme das umgerechnete Bild korrekt einlesen können.

## 6.5 Sequenzaufnahmen

Die *VLab*-Software erlaubt die automatische Aufnahme ganzer Bildfolgen, sogenannter Sequenzen. Natürlich ist es nicht möglich, eine Filmszene Bild für Bild in Echtzeit zu digitalisieren — dafür ist der Amiga einfach nicht schnell genug (siehe auch Abschnitt 6.6). Es werden aber trotzdem beachtliche Bildraten erzielt. Dadurch entstehen Bildfolgen, die an eine Fotokamera mit motorischem Filmtransport erinnern. Der Anwender kann dann entweder gelungene Einzelaufnahmen heraussuchen oder selbst kleine Animationen bzw. „Fotoromane“ herstellen.

Nach der Öffnung des Sequenz-Fensters muß man zunächst den Namen der Bildfolge bestimmen. Dazu klickt man auf „Auswahl“, worauf ein File-Requester erscheint, mit dem sich der Dateipfad und der Grundname der Sequenz festlegen läßt. Für die einzelnen Bilder wird bei der Aufnahme eine vierstellige Nummer an diesen Namen angehängt (Grundname.xxxx).

Danach können wir auswählen, in welchen Betriebsarten die Bilder digitalisiert werden sollen (HiRes/LoRes, Vollbild/Halbbild, Farbe/SW). Diese entsprechen den Modi, die auch bei der einzelnen Digitalisierung zur Verfügung stehen.

Nun stellt man die gewünschte Länge der Bildfolge ein. Eine *VLab*-Sequenz kann aus bis zu 9999 Einzelbildern bestehen. Um die Eingabe zu erleichtern, existiert außer dem Slider-Gadget auch noch ein Cycle-Gadget, mit dem sich der Wertebereich anpassen läßt.

Anschließend sollte man die Dauer der Pause auswählen. Diese bestimmt den zeitlichen Abstand zwischen den Aufnahmen und darf minimal null und maximal 9999 Sekunden betragen. Über ein Cycle-Gadget läßt sich auch hier der Wertebereich umschalten. Natürlich kann die gewünschte Zeitspanne nur dann eingehalten werden, wenn sie länger als die zur Bildspeicherung benötigte Zeit ist. Schnelle und kurze Sequenzen lassen sich deshalb am besten in der RAM-Disk ablegen. Die Pause kann man über das Fenster in Schritten von einer zehntel bzw. einer ganzen Sekunde und per ARexx im 1/50s-Raster einstellen. Beträgt der Zeittakt mindestens zwei Sekunden, wird die *VLab*-Hardware zwischenzeitlich freigegeben, damit z.B. der Monitor weiterlaufen kann. Die Pausenfunktion läßt sich hervorragend zur Zeitrafferaufnahme von Langzeitvorgängen (etwa dem Verhalten einer Blüte) einsetzen.

Es muß jetzt noch der Verkleinerungsmaßstab für unsere Aufnahmen eingestellt werden. Dieser kann von 1:1 (Normalgröße) bis 1:4 variiert werden. Natürlich erreicht man mit kleineren Bildern höhere Bildwechselfrequenzen — die Qualität (Auflösung) ist allerdings geringer. Die Aufnahme von Halbbildern in den Maßstäben 1:2, 1:3 und 1:4 wurde übrigens speziell auf Geschwindigkeit optimiert.

Danach können wir unsere Sequenzaufnahme im gewünschten Augenblick starten. Über die Anzeige „Nächstes Bild“ läßt sich der Vorgang verfolgen. Will man die Aufnahme vorzeitig abbrechen, drückt man auf „Stopp“. Man kann nun entweder durch „Weiter“ mit der nächsten Bildnummer fortfahren oder durch „Start“ wieder mit dem ersten Bild beginnen.

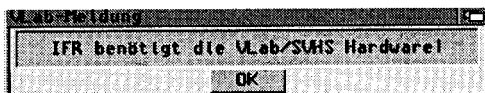
Die Einzelbilder der *VLab*-Sequenzen werden aus Geschwindigkeitsgründen nicht als YUVN-IFF-Datei, sondern in einem internen Format abgespeichert, das nur von *VLab* selbst geladen werden kann. Deshalb müssen die *VLab*-Bildfolgen vor der Weiterbearbeitung mit anderen Programmen umgewandelt werden. Zur automatischen Konvertierung ganzer Sequenzen in das YUVN- oder in ein Amiga-Dateiformat wurden daher Makros zur Verfügung gestellt (siehe Abschnitt 8.4).

## 6.6 Interleaved Frame Recording

### 6.6.1 Allgemeines

*VLab* kann einzelne Bilder zwar in Echtzeit, also in einer 1/25 bzw. 1/50 Sekunde digitalisieren, jedoch in dieser kurzen Zeit nicht in den Amiga einlesen und abspeichern. Die über den Zorro II-Bus maximal erzielbare Datenrate von ungefähr 2 MByte pro Sekunde reicht dazu bei weitem nicht aus. Um 25 Vollbilder pro Sekunde in Farbe einzulesen und wieder abzuspeichern, wären ca. 13 MByte Daten pro Sekunde zu übertragen. Bei dieser Abschätzung ist die Zeit für Verwaltungsaufgaben noch nicht berücksichtigt.

Sollten bisher vollständige Filmsequenzen eingelesen werden, so mußte dies mit einem Einzelbild-fähigen Videorecorder o.ä. Bild für Bild erfolgen. Das neue Verfahren „Interleaved Frame Recording“ (im folgenden kurz IFR genannt) benötigt hingegen keine teure Videoausrüstung mehr, um Filmsequenzen mit hohen Bildwechselfrequenzen zu digitalisieren und abzuspeichern.



IFR arbeitet ausschließlich mit der neuesten VLab-Hardware (SVHS-Version) zusammen. Dabei ist es gleichgültig, ob als Quelle FBAS oder SVHS (YC) benutzt wird. Zusätzlich wird das Vorhandensein der VLab-Softwareversion 4.0 (oder höher) vorausgesetzt.

## 6.6.2 Wie funktioniert IFR?

Wie der Name „Interleaved“ schon vermuten läßt, geschieht die Digitalisierung und das Speichern der Bildfolge in mehreren Durchläufen. Daraus folgt, daß der Videorecorder dieselbe Szene mehrfach hintereinander abspielen muß. Zur exakten Synchronisation wurde in die VLab-Software eine Echtzeit-Bilderkennung eingebaut. Diese wird zunächst dazu eingesetzt, ein sogenanntes Referenzbild (engl. key frame) festzulegen. Dabei handelt es sich eigentlich um zwei direkt aufeinanderfolgende Bilder, die sich deutlich voneinander unterscheiden müssen, z.B. ein Szenenwechsel. VLab merkt sich eine solche Position und kann sie nach Zurückspulen des Videorecorders exakt wiederfinden. Auf diese Weise wird dann bei der eigentlichen Aufnahme diese Stelle mehrfach als Referenz benutzt.

Bei jedem Durchlauf digitalisiert und speichert VLab dann so viele Videobilder wie möglich. Dabei bekommt jedes Bild eine Nummer angehängt, die genau der Position in der fertigen Sequenz entspricht. Der Videorecorder muß dann je nach Frame-Rate mehr oder weniger oft die gleiche Szene abspielen. Die Einlese-Prozedur wird demnach so oft wiederholt, bis VLab sämtliche Bilder digitalisieren und verarbeiten konnte.

## 6.6.3 Die einzelnen Bedienelemente

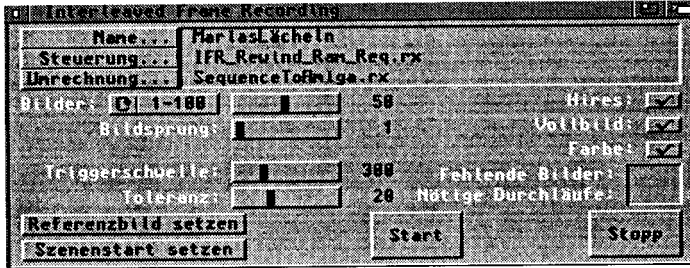


Abbildung 6.1: Das Fenster für **Interleaved-Frame-Recording**

### 6.6.3.1 Bilder

Über das Slider-Gadget wird die Anzahl der Bilder eingestellt, die insgesamt, also in allen Durchläufen zusammen, zu der aufzunehmenden Filmsequenz gehören sollen. Um die Einstellung zu vereinfachen, kann über das Cycle-Gadget der Einstellbereich umgeschaltet werden.

### 6.6.3.2 Bildsprung

Über „Bildsprung“ (engl. frame skip) kann gewählt werden, mit welchem Abstand die einzelnen Bilder digitalisiert und gespeichert werden sollen. Steht Bildsprung auf „1“, wird jedes Vollbild aufgenommen, also 25 Vollbilder pro Sekunde (bei PAL). Sollen nur Halbbilder aufgezeichnet werden, liest das IFR bei „Bildsprung=1“ 25 Einzelbilder pro Sekunde ein und zwar ausschließlich „gerade Halbbilder“ (engl. even frames).

Beispiele:

Bildsprung	Bilder/Sekunde
1	25
2	12.5
5	5
10	2.5
25	1



### **6.6.3.3 Triggerschwelle**

Wie bereits angedeutet, wird zur Synchronisation am besten ein Szenenwechsel benutzt. Je höher der Wert für die Triggerschwelle eingestellt ist, umso größer müssen die Unterschiede zwischen dem Ende der einen und dem Anfang der nächsten Szene sein, damit IFR ein Referenzbild erkennt. Zu Anfang sollte ein Wert von 300 benutzt werden. Dieser Wert ist unter Umständen den Gegebenheiten anzupassen.

### **6.6.3.4 Toleranz**

Weil Videorecorder Fernsehbilder analog speichern, treten bei der Wiedergabe mehr oder weniger starke Abweichungen gegenüber dem Original auf. Neben dem unvermeidlichen Bildrauschen kommen u.a. Änderungen der Bildhelligkeit und Störungen durch schlechtes oder verschmutztes Bandmaterial hinzu.

Um den gewünschten Szenenanfang wiederfinden zu können, muß die Bilderkennung daher mit einer gewissen Toleranz arbeiten. Wie groß diese sein soll, kann über dieses Slider-Gadget eingestellt werden. Wir empfehlen einen Ausgangswert von 20. Man sollte diesen Wert jedoch den eigenen Erfordernissen anpassen. Wird trotz einer recht hohen Triggerschwelle eine Szene auch nach mehreren Versuchen nicht wiedergefunden, sollte der Wert für die Toleranz heraufgesetzt werden.

### **6.6.3.5 Hires, Vollbild, Farbe**

Über diese drei CheckBox-Gadgets kann das gewünschte Bildformat gewählt werden. Dabei sind beliebige Kombinationen möglich.

### **6.6.3.6 TriggerLine**

Über den AREXX-Befehl IFRTRIGGERLINE wird eine Bildschirmzeile gewählt, in deren Bereich VLab die Überprüfung der Bilddaten für die Referenzbild-Erkennung durchführt. Die Zeilennummern beziehen sich auf ein Halbbild und können im Bereich von 1 bis 300 liegen. Je nach Rechenleistung des verwendeten Computers liegt die maximale Zeilennummer aber unter 300. Auf einem Amiga 3000 mit 25 MHz Taktfrequenz liegt die Grenze bei etwa 270. Ein einfacher Amiga mit 68000-Prozessor arbeitet bis zu einer Zeilennummer von etwa 235 (mit A 2620: 265).

Der Standardwert von 70 muß allerdings bei Bedarf verändert werden. Zum Beispiel kann bei einem extremen Breitwandfilm die 70. Bildschirmzeile ständig vollkommen schwarz sein. Werte kleiner als 23 sind nicht sinnvoll, da in diesem Bereich Videotext, Testsignale usw. übertragen werden.

Das Auslesen oder Verändern kann über das Fenster „Befehl ausführen...“, die *VLab*-Shell oder direkt über AReXX erfolgen.

### 6.6.3.7 Name...

Vor einer Aufnahme muß zunächst der Basisname der gewünschten Bildfolge bestimmt werden. Dazu wird das Gadget „Name...“ angeklickt, worauf sich ein File-Requester öffnet, mit dem dann Pfad und Basisname angegeben werden können. An den Basisnamen (z.B. „Test“) werden beim Speichern dann neben einem Punkt die Nummern der einzelnen Bilder angehängt. Nach Aufnahme einer IFR-Sequenz mit 10 Bildern befinden sich in diesem Beispiel daraufhin Bilder mit den Namen „Test.0000“, „Test.0001“ bis „Test.0009“ in der RAM-Disk. Bei den Nummern handelt es sich um vierstellige Dezimalzahlen; die Zählweise startet mit 0.

### 6.6.3.8 Steuerung...

Nach Anklicken des Gadgets „Steuerung...“ kann ein Makro gewählt werden, daß immer dann aufgerufen wird, wenn der nächste Durchlauf gestartet, also der Videorecorder zurückgespult werden muß. Wie von *VLab* gewohnt, kann es sich um ein *VLab*-Makro, ein AReXX- oder ein normales Shell-Skript handeln. Dieses Makro wird auch am Ende einer erfolgten Aufnahme noch einmal aufgerufen, um auch die restlichen Bilder zu kopieren oder zu konvertieren.

Wird kein Makro-Name angegeben, erscheint lediglich ein Requester, der zum Rückspulen auffordert (nicht jedoch am Ende).

## Ausnutzung der RAM-Disk

Um die Anzahl der IFR-Durchläufe niedrig zu halten und damit Zeit zu sparen, sollte zunächst in die RAM-Disk gespeichert werden. Nach jedem Durchlauf oder wenn die RAM-Disk voll ist, wird das Rewind-Makro aufgerufen, daß die Dateien dann auf die Festplatte kopiert und in der RAM-Disk löscht.

Beispiel:

```
/* IFR-Rewind_RAM */
OPTIONS RESULTS

ADDRESS COMMAND 'Copy RAM:Test.#? DH0:'
ADDRESS COMMAND 'Delete RAM:Test.#?'
ADDRESS VLAB    'REQUEST TITLE IFR GADGETS OK "Rewind!"'
```

## Benutzung von AirLink

Eine Verbesserung des zuvor beschriebenen Makros läßt sich erreichen, wenn AirLink zum Einsatz kommt. Zur Erinnerung: AirLink ist eine lernbare Infrarot-Fernbedienung für den Amiga. So reicht dann ein ARexx-Befehl aus, um den Videorecorder für den nächsten Durchlauf zurückzuspulen. Dadurch wird eine völlige Automatisierung des IFR erreicht!

Beispiel:

```
/* IFR-Rewind_RAM_AIR */
OPTIONS RESULTS

ADDRESS COMMAND 'Copy RAM:Test.#? DH0:'
ADDRESS COMMAND 'Delete RAM:Test.#?'
ADDRESS AIR      'OUTPUT IR< ACS >'
```

## Konvertierung

Wer die Sequenz in einem anderen Format abgelegt haben möchte, z.B. HAM, sollte die Konvertierung am besten im Rewind-Makro in der RAM-Disk ausführen. Das spart eine Menge Zeit und Festplatten-Platz. Vor Kopieren der Daten auf die Festplatte muß dann die Konvertierung aller sich in der RAM-Disk befindlichen Bild-Dateien erfolgen. Denken Sie bei der Erstellung des Makros daran, daß die einzelnen Dateinamen unterschiedliche, nicht aufeinanderfolgende Nummern am Ende ihres Namen tragen!

### 6.6.3.9 Umrechnung...

Das Umrechnen-Makro wird nach erfolgter Aufnahme der gesamten Sequenz aufgerufen, nicht jedoch nach jedem Durchlauf. Es kann daher zur Konvertierung der gesamten Sequenz in ein anderes Format oder zur Erstellung einer Animationsdatei eingesetzt werden.

## 6.6.4 Durchführung einer Aufnahme

### 6.6.4.1 Festlegen des Referenzbildes

Wie bereits beschrieben, benötigt IFR ein Referenzbild, um bei jedem Durchlauf auch exakt die richtigen Bilder zu digitalisieren. Als Referenzbild eignen sich alle Stellen, an denen der Unterschied von einem zum nächsten Bild recht groß ist. Wie groß die Differenz sein muß, kann über „Triggerschwelle“ eingestellt werden. Je höher der Wert,

um so größer müssen die Bildunterschiede sein. Zu Anfang sollte mit einem Wert um 300 experimentiert werden.

Starten Sie die Wiedergabe des Videorecorders einige Sekunden vor der gewünschten Szene und warten Sie, bis sich das Bild stabilisiert hat. Wenn Sie nun „Referenzbild setzen“ anklicken, analysiert *VLab* ab diesem Zeitpunkt laufend das Videosignal. Sobald ein Referenzbild erkannt wurde, merkt sich *VLab* diesen Bildwechsel für die spätere Bilderkennung.

Sollte vor Beginn der aufzunehmenden Szene kein Referenzbild gefunden werden, müssen Sie weiter zurückspulen und es nochmal versuchen. Sie können natürlich auch die Triggerschwelle verkleinern, und damit auch kleinere Bildänderungen als Referenzbild akzeptieren. Dadurch verringert sich allerdings die Sicherheit bei der Wiedererkennung. Bei sehr kleinen Werten sollte auch der Wert für „Toleranz“ recht klein gewählt werden. Dazu ist dann aber eine qualitativ höherwertige Quelle notwendig.

Wurde ein geeignetes Referenzbild gefunden, können Sie auf „Start“ klicken, den Recorder wieder bis einige Sekunden vor das Referenzbild zurückspulen und mit der eigentlichen Aufnahme beginnen. Diese beginnt genau ab dem gefundenen Referenzbild.

#### **6.6.4.2 Setzen eines Szenenanfanges**

Häufig kommt es vor, daß längere Zeit vor der aufzunehmenden Szene kein Referenzbild gefunden wird. Um das Digitalisieren und Abspeichern unnötiger Daten zu vermeiden, kann ein verzögerter Szenenstart definiert werden. Bestimmen Sie dazu zunächst ein Referenzbild, wie zuvor beschrieben. Wurde es gefunden, lassen Sie den Videorecorder bis zum Anfang der gewünschten Szene weiterlaufen. Zu diesem Zeitpunkt müssen Sie dann auf „Szenenstart setzen“ klicken.

Danach kann mit der eigentlichen Aufnahme begonnen werden, die jetzt nicht schon am Referenzbild, sondern erst an der Stelle beginnt, an der Sie den Szenenstart bestimmt haben. Beachten Sie bitte, daß zur Synchronisation trotzdem bis kurz vor das Referenzbild zurückgespult werden muß. Die Daten zwischen Referenzbild und Szenenstart werden bei der Aufnahme völlig ignoriert.

**Achtung:** Verändern Sie während der Festlegung eines Szenenstarts keine Einstellungen an Ihrem Videorecorder!

#### **6.6.4.3 Aufnahme einer Szene**

Die Festlegung eines Referenzbildes und evtl. eines Szenenstarts muß an dieser Stelle bereits durchgeführt worden sein. Zur Aufnahme klicken Sie nun das Gadget „Start“ an. Sie werden dann zunächst aufgefordert, bis vor das Referenzbild zurückzuspuhlen. Dabei

sollte man nicht zu knapp vor dem Referenzbild beginnen, da ein Videorecorder einige Zeit benötigt, um nach dem Start ein vollkommen stabiles Bild zu liefern. Nachdem dies geschehen ist, starten Sie Ihren Recorder und klicken den Requester weg, sobald sich das Bild stabilisiert hat. VLab sucht daraufhin das zuvor gesetzte Referenzbild. Wurde kein Szenenstart gesetzt, beginnt die Aufnahme direkt an dem Referenzbild, andernfalls wird bis zum Szenenstart abgewartet.

Sollte VLab das Referenzbild nicht wiedererkennen, spulen Sie den Recorder zurück und versuchen es zunächst noch einmal! Wird dadurch kein Erfolg erzielt, setzen Sie den Wert von „Toleranz“ höher und probieren es erneut. Sollte alles nichts helfen, wird es notwendig, ein neues Referenzbild zu bestimmen. Dieser Fall kann z.B. eintreten, wenn gerade bei der Definition des Referenzbildes eine Bildstörung auftrat (Tracking-Fehler, Bildstörung durch Elektrogerät, ...).

Am Ende der Szene angelangt, fordert VLab wiederholt zum Rückspulen auf, bzw. ruft das Rückspul-Makro auf, bis alle Bilder der Szene digitalisiert und abgespeichert sind. Wieviele Bilder noch fehlen und wieviele Durchläufe voraussichtlich noch benötigt werden, wird im Kontrollfenster angezeigt.

**Achtung: Während eines Durchlaufs dürfen keine Einstellungen am Videorecorder verändert werden!**

#### **6.6.4.4 Fortsetzung einer Aufnahme**

Über den Menüpunkt „Steuern/Film/Weiter“ kann eine IFR-Aufnahme fortgesetzt werden, sofern VLab zwischenzeitlich nicht beendet wurde. Dabei wird das zuvor eingestellte Referenzbild benutzt.

Falls eine zuvor durchgeführte Aufnahme vorzeitig abgebrochen wurde, können die fehlenden Bilder auf diese Weise nachträglich eingelesen werden.

Wenn an die zuvor vollständig eingelesene IFR-Sequenz noch weitere Bilder angehängt werden sollen, braucht nur die Anzahl Bilder auf den neuen Wert gesetzt werden. Wurden zuvor z.B. 50 Bilder eingelesen und sollen noch 20 weitere angehängt werden, muß die Bildanzahl auf 70 erhöht werden.

Sollte eine Kontrolle der eingelesenen Bilder ergeben, daß einzelne Bilder Fehler aufweisen, können diese nachträglich erneut eingelesen werden. Dazu existiert allerdings keine Bedienoberfläche. Sind z.B. von einer 500 Bilder langen IFR-Sequenz die Bilder 13, 274 und 388 fehlerhaft, müßten folgende Befehle über ARexx oder das Fenster „Befehl ausführen“ gegeben werden, bevor die Aufnahme fortgesetzt werden kann:

```
IFRFRAME SET ALL
IFRFRAME CLEAR 13
IFRFRAME CLEAR 274
IFRFRAME CLEAR 388
```

#### 6.6.4.5 Das Ausgabe-Fenster

Während der Erstellung einer IFR-Sequenz erfolgen in einem eigenen Fenster zusätzliche Ausgaben. Größe und Position dieses Fensters werden im Gegensatz zu allen anderen *VLab*-Fenstern nicht automatisch gespeichert. Änderungen können nur über den Befehl „CHANGEHELPWINDOW“ erfolgen.

#### 6.6.5 Erstellen einer Animation

Aus den mit IFR aufgenommenen Einzelbildern kann mit anderen Programmen eine Animation erzeugt werden. Je nach verwendeter Software müssen die Bilder zuvor aber noch in ein anderes Format gewandelt werden.

Um die Erstellung von Animationen zu erleichtern, speichert IFR in den Dateien eine Zeitinformation mit ab (Time-Code). So kann auch nachträglich die Bildnummer und exakte zeitliche Position festgestellt werden.

#### 6.6.6 Tips zu IFR-Problemen

Da es sich bei IFR um ein besonders zeitkritisches Verfahren handelt, sollten währenddessen keine Operationen ausgeführt werden, die das System extrem belasten. Besonders Operationen, die stark mit Interrupts arbeiten, oder diese über einen längeren Zeitraum hinweg abschalten, sollten vermieden werden. Dazu gehört z.B. die Datenübertragung mit einem Modem. Auch einige Festplattencontroller sperren über längere Zeit die Interrupts oder blockieren den Bus mit unnötig langen DMA-Zugriffen. In diesem Fall müssen die Bilder in der RAM-Disk zwischengespeichert und von Zeit zu Zeit auf die Festplatte kopiert werden, z.B. im Rewind-Makro. Ein Umschalten oder Verschieben von Screens kann zu Bildstörungen führen und sollte ebenfalls vermieden werden.

Die zeitkritische Arbeitsweise von IFR macht es auch notwendig, daß die *VLab*-Monitorfunktion zeitweise angehalten werden muß. Der Einsatz eines zusätzlichen Kontrollmonitors an der Videoquelle ist daher empfehlenswert.

- die optimale Anpassung an eigene Graphikhardware (siehe 6.8.3),
  - die optische Verbesserung bei schlechter Signalqualität,
  - die künstlerische Verfremdung durch extreme Einstellungen.
- Die von *Lab* digitalisierten Videobilder besitzen in der Regel die richtigen, natürlichen Farböne. Dennoch können Situationen entstehen, in denen der Anwender die Erzeugung der Farben beeinflussen möchte. Dazu zählen unter anderem:

## 6.7 Die Farbkorrektur

**Beim Start von IFR bleibt der Rechner (nahezu) stehen.**  
Auf der *Lab/YC*-Hardware muß ein Bauteil getauscht werden. Bitte wenden Sie sich für den kostenlosen Austausch direkt an MacroSystem!

- Ihre Hardware hat thermische Effekte. Wenden Sie sich für den kostenlosen Austausch eines Bauteils bitte direkt an MacroSystem!
- Speichern Sie nicht direkt auf Festplatte, da der Controller stören könnte. Benutzen Sie statt dessen die RAM-Disk (siehe Zurückspulen).
- Kontrollieren Sie Ihren Videorecorder auf Gleichlauf- oder Tracking-Fehler.
- Stellen sie die Triggerschwelle höher ein und benutzen gegebenenfalls ein anderes Referenzbild.
- Verkleinern Sie den Wert für die Toleranz.
- Überprüfen Sie den Wert für "TRIGGERLINE". Er sollte etwa bei 70 liegen.
- Erhöhen Sie den Wert für "Toleranz".
- Reduzieren Sie den Wert für "Triggerschwelle".

**Es wird kein Referenzbild erkannt.**

Die Einstellung der Farbreger wirkt sich sowohl auf die Umrechnung in Amiga-Grafiken, als auch auf den Export von 24 Bit-Bildern aus. Dagegen wird bei der Vorschau die Farbkorrektur nicht verwendet. Eine Ausnahme besteht jedoch in der Darstellung auf einer 24 Bit-Grafikkarte.

In der Neutralstellung wird das digitalisierte Videobild nicht verfremdet. Diese läßt sich mit dem Gadget „Normal“ jederzeit wieder herstellen. Im einzelnen stehen dem Benutzer Einstellregler für folgende Parameter zur Verfügung:

### **6.7.1 Farbe**

Dieser Slider beeinflusst den Farbanteil des Bildes. Einen entsprechenden Regler gibt es an fast jedem Farbfernsehgerät. Dort heißt er in der Regel „Farbsättigung“. In Normalstellung (100%) bleibt der Farbanteil im Bild voll erhalten. Bei 0% werden dem Bild alle Farbinformationen entzogen, es bleibt ein reines Schwarzweißbild übrig. Durch die freie Wahl von Zwischenwerten kann jedes beliebige Verhältnis eingestellt werden. Jenseits von 100% wird der Farbanteil angehoben. Damit ist natürlich kein Einfärben eines Schwarzweißbildes möglich, da aber z.B. viele alte Spielfilme sehr blasse Farben haben, ist eine Anhebung des Farbanteils durchaus interessant. In allen Fällen bleibt die Helligkeit des Bildes unverändert.

### **6.7.2 Rot,Grün,Blau**

Ein Bild wird auf einem Monitor oder Fernseher immer aus den drei additiven Grundfarben Rot, Grün und Blau zusammengesetzt. Obwohl *VLab* (wie natürlich auch Fernseher, Videorecorder, usw.) intern mit YUV arbeitet, wird vor der Ausgabe das YUV-Signal wieder in RGB zurückgewandelt. Bei *VLab* geschieht dies während der Umwandlung eines YUV-Bildes in ein Amiga-Bild oder bei der Konvertierung in 24 Bit.

Ein ungleichmäßiges Verschieben der Slider Rot, Grün oder Blau von der Nullstellung (100%), beeinflusst die Umrechnung so, daß keine gleichförmige Verteilung der Grundfarben mehr vorliegt. Dadurch lassen sich Farbstiche verhindern oder ein fehlerhafter Weißabgleich bei einer Kamera korrigieren.

Steht einer der Farbreger auf 0%, so tritt diese Farbkomponente im konvertierten Bild überhaupt nicht mehr auf. Bei einer Stellung über 100% wird die entsprechende Grundfarbe verstärkt.



### 6.7.3 Kontrast

Auch ein Kontrastregler dürfte vom Fernsehgerät her bekannt sein. Wird der Kontrast erhöht ( $> 100\%$ ), werden dunkle Bildbereiche noch dunkler und helle Bereiche noch heller. Dadurch werden Details im mittleren Helligkeitsbereich deutlicher, dafür gehen bei starker Anhebung häufig unteren und oberen Helligkeitsbereich Informationen verloren. Bei einer Absenkung des Kontrasts wird der Bereich von dunkel bis hell verkleinert, das Bild wird zunehmend mittelgrau und die Konturen werden schwächer.

### 6.7.4 Helligkeit

Ein Regler zur Einstellung der Helligkeit eines Bildes ist ebenfalls an jedem Fernsehgerät oder Monitor vorhanden. Obwohl man an einem solchen Gerät häufig keinen Unterschied in der Wirkung des Kontrast- und des Helligkeitsreglers sieht, ist sehr wohl ein Unterschied vorhanden.

Bei einer Veränderung der Bildhelligkeit werden alle vorkommenden Helligkeitswerte gleichmäßig angehoben oder abgesenkt. Eine Veränderung der Helligkeitsauflösung findet nicht statt. Lediglich bei dunklen Bereichen bei der Absenkung der Helligkeit und bei hellen Bereichen bei der Anhebung gehen einige Bildinformationen verloren.

### 6.7.5 Gamma

Die Gamma-Korrektur, versucht die Vorteile der Kontrast- und der Helligkeitskorrektur in einer Funktion zu vereinen. Das Bild kann aufgehellt oder abgedunkelt werden, ohne daß zu viele Informationen verloren gehen. Das geschieht durch den Einsatz einer nichtlinearen Übertragungsfunktion.

Diese Funktion läßt schwarze Gebiete schwarz und weiße weiß. Der Bereich dazwischen wird dagegen beeinflußt und zwar werden bei einer Anhebung (positiver Gamma-Wert) besonders die dunkleren Bereiche aufgehellt, wogegen die helleren Bereiche nur wenig verändert werden. Andererseits werden bei einer Absenkung (negativer Gamma-Wert) besonders die helleren Bereiche abgedunkelt, die dunkleren Bereiche werden nur wenig dunkler. Es ist zu beachten, daß eine positive Gamma-Korrektur leider auch das Bildrauschen erhöht.

Die Positionen der Schieberegler können auch über den AREXX-Befehl „COLORVALUE“ gesetzt und mit „GETCOLORVALUE“ gelesen werden.

Ein Tip: Probieren Sie die Auswirkungen der sieben Regler einfach aus — es lassen sich erstaunliche Effekte, aber unter Umständen auch ernsthafte Verbesserungen erreichen.

## 6.8 Probleme mit Amiga IFF-ILBM-Dateien

Eigentlich sollten diese Erklärungen nicht notwendig sein, leider haben aber einige bekannte Programme Schwierigkeiten mit den von *VLab* im IFF-ILBM-Format gespeicherten Bildern.

### 6.8.1 Kompression

Einige Anwendungen können *nur* komprimierte Dateien einlesen (siehe Seite 71).

### 6.8.2 Bildbreite

*VLab* liefert IFF-Bilder mit nahezu beliebiger Breite und Höhe. Einige Programme kommen aber z.B. mit Bildbreiten, die kein gerades Vielfaches von acht sind, nicht zurecht und machen das Bild einfach breiter als es ist. Am rechten Bildschirmrand wird dann meist zusätzlich ein schmaler Streifen in der Hintergrundfarbe angezeigt, der dann in der Regel auch noch mit abgespeichert wird. Hierbei wurden die IFF-Richtlinien nicht beachtet. Es handelt sich nicht um einen Fehler von *VLab*! Ist man gezwungen, mit solchen Programmen zu arbeiten, definiert man am besten die *VLab*-Videoquellen gleich so, daß die Anzahl der Bildpunkte in X-Richtung durch 16 teilbar ist.

### 6.8.3 Farbpalette

Die Farbpalette wird von *VLab* (anders als bei vielen anderen Programmen) auf maximalen Kontrast optimiert. Diese Methode wird in den IFF-Richtlinien von Electronic Arts und Commodore empfohlen. Daher können beim direkten Vergleich von zwei Bildern, die von *VLab* und einem anderen Programm konvertiert wurden, evtl. leichte Unterschiede festgestellt werden. Das von *VLab* benutzte Verfahren bietet besonders beim Übertragen der Bilder auf andere Computer mit anderen Grafikfähigkeiten enorme Vorteile. Unterschiedliche Umrechnungsergebnisse können ihre Ursache natürlich auch in einem anderen Verfahren zur Bildkonvertierung haben.

Es ist bemerkenswert, wie unterschiedlich dieselbe Amiga-Grafik auf verschiedenen Bildschirmen und Grafikkarten aussehen kann. Häufig wirken die Bilder auf Computermonitoren etwas unnatürlicher als auf Fernsehapparaten, was auf eine andere Auslegung der Bildröhrentechnik zurückzuführen ist. Doch auch zwischen zwei Computermonitoren kann derselbe Farbton signifikant differieren. Daher sollten Sie die vielfältigen Möglichkeiten der *VLab*-Farbkorrektur (siehe Abschnitt 6.7) anwenden, wenn Sie eine Grafik optimal auf Ihr System anpassen wollen.

# Kapitel 7

## Etwas Videotechnik

### 7.1 Das FBAS-Signal

Die *VLab*-Hardware benötigt das zu digitalisierende Videobild in Form eines sogenannten FBAS-Signals (englisch CVBS). In diesem Kombinationssignal sind alle Informationen kodiert, die zur Darstellung eines farbigen Fernsehbildes notwendig sind. Im einzelnen sind dies:

- Die Farbinformation (Color):  
Diese setzt sich aus zwei Farbdifferenzsignalen (Chrominanz) zusammen und liegt zusätzlich zur Bildinformation vor. Aus den beiden Farbkoordinaten werden mit Hilfe der Helligkeitswerte erst im Fernsehempfänger die Rot/Grün/Blau-Anteile für die Bildröhre gebildet. Dadurch konnte das Farbfernsehen kompatibel zu Schwarzweiß-Fernsehgeräten gehalten werden.
- Die Bildinformation (Video):  
Sie enthält nur die Helligkeitsamplitude (Luminanz) des Videobildes. Im Schwarzweiß-Betrieb ist dies der einzige Bildinhalt. Beim Farbfernsehen kommen noch zwei Farbkoordinaten hinzu (siehe oben).
- Die Austastlücken (Blanking):  
Diese sind Zeitbereiche zwischen den Halbbildern und zwischen den einzelnen Bildzeilen, in denen keine Bilddaten übertragen werden. Sie erlauben es, den Elektronenstrahl der Fernröhre in die Startposition für das nächste Teilbild bzw. die nächste Zeile zurückzuführen. In der vertikalen Austastlücke werden allerdings, für den Betrachter unsichtbar, zusätzliche Informationen (wie z.B. Videotext, VPS-Label und Testsignale) gesendet.

- Die Synchronisationssignale (Syncs):

Sie dienen dazu, den Aufbau des Videobildes richtig zu takten. Dabei kennzeichnet das vertikale Synchronsignal den Beginn eines neuen Halbbildes, während das horizontale Synchronsignal den Start der nächsten Bildschirmzeile einleitet. Beim Farbfernsehen existiert außerdem noch ein Farbträgersynchronsignal (Burst).

Diese Informationen werden also (in unterschiedlichster Weise kodiert) zum FBAS-Signal zusammengefaßt und werden, gemeinsam mit den zugehörigen Audio-Signalen, einem Träger wesentlich höherer Frequenz aufmoduliert, einen Fernsehkanal. Bei der Ausstrahlung (egal, ob über Antenne, Kabel oder Satellit) liegen jedoch viele Kanäle gleichzeitig als Hochfrequenzgemisch vor. Deshalb braucht man den Empfänger (Tuner) eines Fernsehgerätes oder Videorecorders, um das gewünschte Programm für *VLab* zu separieren. Wird das Videosignal über eine „Y/C“-Leitung übertragen (S-VHS, Hi8), dann existiert im Kabel für den Farbanteil eine eigene Ader. Durch die getrennte Signalführung können gegenseitige Störeinflüsse zwischen der Farbinformation und dem Schwarzweiß-Fernsehbild vermieden werden.

## 7.2 Was bedeutet eigentlich „YUV“?

YUV (sprich Wei-Ju-Wie) ist das weltweit gebräuchliche Kodierungsformat für farbige Videobilder. Jeder Videorecorder, jeder Fernseher benutzt es. Im YUV-Format werden die Helligkeitswerte (Luminanz, Y) und die Farbkoordinaten (Chrominanz, U und V)<sup>1</sup> getrennt verwaltet (siehe Abschnitt 7.1). Als Folge davon kann man etwa bei einem Videosignal nur jeden vierten Bildpunkt mit einer neuen Farbe belegen, die dazwischen liegenden Pixel werden ausschließlich mit Helligkeitsänderungen versehen. Ergebnis ist ein bekanntermaßen lebensechtes und dennoch kompaktes Bild.

Tatsächlich erreichen solche YUV-Bilder nicht die volle Qualität von 24 Bit-Grafiken — im 24 Bit-Format kann jeder einzelne Punkt eine beliebige Farbe aus 16,7 Millionen darstellen. Natürlich gibt es Digitizer, die 24 Bits speichern, man muß aber die Frage nach dem „Wozu?“ stellen. Denn besser als die anliegende Videoquelle zu sein, wirkt sich zwar im Preis und in der Geschwindigkeit, sowie im RAM-Bedarf negativ aus, verbessert aber keinesfalls die schließlich erzielbare Bildqualität.

*VLab* wurde so konzipiert, daß der volle Informationsgehalt, den Videoquellen (Fernsehempfänger, Kabeltuner, Videorecorder, Kamera etc.) bieten, digitalisiert und gespeichert werden kann. Deshalb erscheint uns ein echter 24 Bit-Digitizer für die meisten Videoanwendungen überdimensioniert.

<sup>1</sup>NTSC verwendet ähnliche Farbkoordinaten, die I und Q genannt werden. Der *VLab*-Chipsatz muß daher vom Anwender auf den NTSC-Betrieb umgeschaltet werden.

Dennoch unterstützt die *VLab*-Software auch das 24 Bit-Format, weil es eben Produkte wie die Grafikkarten *Retina* und „Harlequin“ gibt, welche die volle 24 Bit-Farbtiefe darstellen bzw. verarbeiten können. So kann man mit Programmen wie TVPaint oder VDPaint auch YUV-Bilder auf 24 Bit-Ebene nachbearbeiten.

### 7.3 Braucht man für VLab eine Grafikkarte?

Um *VLab*-Bilder darstellen zu können, wird in der Regel keine besondere Grafikkarte benötigt. Sicher sehen die Bilder auf 24 Bit-Karten besser aus — aber schließlich ist der Amiga ein Computer, der gerade wegen seiner serienmäßig sehr guten Grafik berühmt geworden ist.

Die Umrechnung in typische Amiga-Bildformate kann, wie bereits beschrieben, direkt durch die *VLab*-Software geschehen. Oft wird man ein Beibehalten der Amiga-Auflösungen auch wegen des geringeren Speicherbedarfs und der vielfältigen Verwendungsmöglichkeiten (Dia-Show-Disk etc.) wünschen.

Zur Wiedergabe von Videobildern sind der bekannte HAM-Modus (4096 Farben bei  $360 \times 580$  Pixeln) sowie dessen Erweiterungen auf dem Amiga 4000 (bis 262144 Farben bei  $720 \times 580$  Pixeln) am besten geeignet.

Leider bieten diese speziellen Video-Betriebsarten nicht die Möglichkeit, jeden Bildpunkt unabhängig von seinen Nachbarn zu verändern. Die Weiterverarbeitung von HAM-Bildern mit leistungsstarken Zeichenprogrammen wird dadurch stark eingeschränkt.

Ist der Anwender also auf die professionelle Bearbeitung des digitalisierten Bildmaterials angewiesen (Collage, Retusche, etc.), dann ist ihm die Anschaffung einer 24 Bit-Grafikkarte anzuraten. Ein reiner Bildspeicher (Framebuffer) ist jedoch ungeeignet, weil auf dessen RAM nur recht langsam zugegriffen werden kann.

Natürlich muß dazu auch die entsprechende Software eingekauft werden. In praxi benötigt man ein Bildbearbeitungsprogramm (wie ADPro, ImageMaster, Morph Plus) und ein 24 Bit-Zeichenprogramm (wie TVPaint, VDPaint).

Wir empfehlen die MacroSystem 24 Bit-Grafikkarte *Retina*, die komplett mit einer Workbench-Emulation und dem o.g. VDPaint geliefert wird. Diese echte Amiga-Grafikkarte bietet ein hervorragendes Preis-Leistungs-Verhältnis und wird natürlich von *VLab* voll unterstützt.

Sollen Genlock-Möglichkeiten eingesetzt werden, müssen die fertig bearbeiteten Bilder z.Zt. noch auf dem Amiga 4000 oder einem speziellen Genlock-fähigen Framebuffer (z.B. Harlequin) ausgegeben werden.

## 7.4 Wird ein „Time-Base-Corrector“ benötigt?

Unter einer „Time-Base-Correction“ (Zeitbasiskorrektur) versteht man ein Gerät, welches vielen Digitalisierern vorgeschaltet werden muß, um direkt von Videorecordern einlesen zu können.

Videorecorder (aber auch andere Videoquellen) haben große Probleme mit Zeitschwankungen im Videosignal, weil die Bandlaufgeschwindigkeit nicht genau genug geregelt werden kann. Deshalb besitzen viele Fernseher einen speziellen Programmplatz, der dieser Tatsache Rechnung trägt.

Für viele Video-Digitizer stellt sich das Problem sogar so verschärft, daß das Einlesen direkt vom Videorecorder nur durch zwischengeschaltete sogenannte Time-Base-Corrector-Schaltungen möglich ist. Diese TBC-Geräte liegen preislich zumeist im Bereich von mehreren tausend Mark.

Bei *VLab* wurde dieses Problem sehr elegant gelöst: Falls man einen Videorecorder als Quelle benutzt, aktiviert man einfach den Menüpunkt „Videorecorder“ in der Software. Dadurch wird ein spezieller Schaltkreis im *VLab*-Chipsatz aktiviert, so daß *VLab* prinzipiell völlig problemlos mit Videorecordern zusammenläuft.

Eine teure externe Zeitbasiskorrektur ist deshalb für den Betrieb von *VLab* nicht notwendig.

## 7.5 Was wird unter „S-VHS“ verstanden?

Beim Stichwort „S-VHS“ muß zunächst eine häufige Verwechslung geklärt werden. Es gibt nämlich:

1. die Bezeichnung S-VHS für Videoquellen mit S-VHS Bildqualität
2. den Anschluß für S-VHS-Verbindungskabel.

Zur Nutzung der Vorteile eines S-VHS-Videoerätes muß nicht unbedingt auch der S-VHS-Anschluß verwendet werden.

Eine S-VHS-Videoquelle hat im Gegensatz zu normalen VHS-Geräten eine horizontale Auflösung von über 700 Bildpunkten (VHS: 380 Pixel) und ist damit für *VLab* geradezu prädestiniert. Diese Auflösung bietet ein S-VHS-Gerät allerdings auch über den (stets vorhandenen) FBAS-Ausgang.

Der S-VHS-Ausgang trennt das Helligkeitssignal (Y) von der Farbinformation (U+V) physikalisch durch die Verwendung verschiedener Leitungen. Die Anzahl der Pixel (Videobandbreite) bleibt dabei übertragungstechnisch gleich. Durch diesen Trick werden

jedoch die sogenannten Cross-Color- bzw. Cross-Luminance-Effekte vermieden. Solche Effekte würden etwa dann entstehen, wenn ein Nachrichtensprecher ein kariertes Jacket trägt — es bilden sich ungewollte Farbsäume, die z.B. eine gute S-VHS-Kamera, wenn sie direkt über ein S-VHS-Kabel angeschlossen wäre, nicht erzeugen würde.

Die etwas höhere Qualität eines digitalisierten Bildes würde jedoch nur von Anwendern von 24 Bit-Grafikkarten bemerkt werden. Die Verwendung einer S-VHS-Kamera bzw. eines S-VHS-Recorders erscheint dagegen sehr sinnvoll, da die Schärfe (Auflösung) des Bildes bei Verwendung eines solchen Gerätes erheblich besser ist.

## 7.6 Die Hardware-Konzeption von VLab

*VLab* wurde unter Verwendung eines Chip-Satzes konzipiert, der normalerweise in sehr teuren Marken-Fernsehern mit 100 Hz-Technik eingesetzt wird. In diesen Chip-Satz mußten unzählige Mann-Jahre Entwicklungszeit investiert werden — logisch, denn niemand würde soviel Geld für einen Fernseher mit schlechter Bildqualität ausgeben. Die Vorteile der Benutzung eines kommerziellen Video-Chip-Sets sind vielfältig:

- Der Preis für *VLab* konnte durch den in großen Stückzahlen gebauten Chip-Satz außergewöhnlich niedrig gehalten werden.
- Durch die hohe Ausgereiftheit sind keine Probleme mit Videoquellen, Synchronisation etc. zu befürchten — ein Thema, das Anwendern von herkömmlichen Geräten sicher ein Begriff ist.
- Im Gegensatz zu vielen bisherigen Geräten ist weder eine „Time-Base Correction“ noch ein RGB-Splitter notwendig, denn kein Käufer würde so etwas bei einem Fernseher akzeptieren...
- Es können Videobilder in unterschiedlichen Fernsehnormen (PAL, NTSC) eingelesen werden.

Das Bild wird also von diesem Chip-Satz umgewandelt und dann in speziellen Video-RAMs auf der *VLab*-Platine zwischengespeichert. Von dort wird es mit 16 Bit Busbreite (ohne Wait-States) in den Systemspeicher des Amiga übertragen.

Verständlicherweise kann die an den Parallelport angeschlossene externe Version *VLab/par* nur langsamer einlesen. Durch eine Hardwareskalierung können jedoch größere Bilder (wie der Monitor) wesentlich schneller in den Rechner gebracht werden als bei Konkurrenzprodukten.

Eine Besonderheit der *VLab*-Hardware besteht im Vorhandensein weiterer Eingangsbuchsen, die es ermöglichen, mehrere Videoquellen gleichzeitig angeschlossen zu halten und so ohne großes „Fummeln“ per Software umschalten zu können.

## 7.7 Das PAL-System

Die in Europa verbreitete PAL-Fernsehnorm basiert auf der Übertragung von 25 Vollbildern zu je 625 Zeilen pro Sekunde. Diese Vollbilder werden in zwei Halbbilder aufgeteilt, welche nacheinander mit einer Halbbildfrequenz von 50 Hz übertragen werden. Die Zusammensetzung des Fernsehsignals wurde bereits in Abschnitt 7.1 beschrieben.

### 7.7.1 Die Schwarzweiß (Y)-Auflösung des PAL-Systems

Die vertikale Auflösung eines Vollbildes beträgt, wie oben gesagt, 625 Zeilen. Bei der Definition des PAL-Standards legte man ein Verhältnis der Bildbreite zur Bildhöhe von 4:3 fest. Weiterhin sollten die Pixel nahezu quadratisch sein. Daraus ergibt sich eine horizontale Auflösung von

$$625 \cdot \frac{4}{3} = 833 \text{ Pixeln.}$$

Ein Bild enthält damit  $625 \cdot 833 \approx 520.000$  Bildpunkte. Pro Sekunde werden 25 Vollbilder, also  $520.000 \cdot 25 = 13$  Mio. Bildpunkte übertragen. Um die höchste vorkommende Frequenz zu ermitteln, nimmt man an, daß die Pixel einer Zeile abwechselnd hell und dunkel sind. Daraus ergibt sich eine Frequenz von  $13/2 = 6,5$  MHz.

Bei der Übertragung eines Fernsehbildes über Antenne wird diese Frequenz auf 5 MHz begrenzt, da nur selten solch hohe Auflösungen vorkommen und man dort leichte Unschärfen in Kauf nimmt, um die benötigte Übertragungsbandbreite zu verringern. Bei 5 MHz erhalten wir eine horizontale Auflösung von ca. 640 Bildpunkten. Der VLab-Digitizer mit seiner horizontalen Auflösung von 720 Punkten erreicht zwar nicht ganz das theoretische Maximum, übertrifft aber die Auflösung eines Fernsehbildes deutlich.

Arbeiten Sie mit einem VHS-Videorecorder, ist der Qualitätsunterschied noch drastischer. Dort wird das Videosignal auf 3 MHz begrenzt. Die resultierende horizontale Auflösung beträgt gerade noch 380 Bildpunkte.

### 7.7.2 Die Farbkodierung im PAL-System

Anstatt wie in der Computertechnik ein RGB-Signal zu übertragen, hat man sich bei der Videotechnik (aus Kompatibilitätsgründen zu vorhandenen Schwarzweißempfängern) entschlossen, zusätzlich zu dem Leuchtdichtesignal ein Farbsignal zu übertragen, welches den Farbton und die Farbsättigung der einzelnen Farben kodiert enthält.



### 7.7.2.1 Das Farbart (Chrominanz)-Signal

Das Farbkordinatensystem nach DIN 5033 besteht aus einer Art Dreieck, mit den Farben Blau, Grün und Rot als Eckpunkten. Am Rand dieses Dreiecks verlaufen die vollen Farben, diese werden als Farbtöne bezeichnet. Zur Mitte hin werden die Farben blasser bis hin zum Weiß. Für die Videotechnik formt man dieses Dreieck zu einem Kreis um, so als ob der Rand des Dreiecks ein Gummiband wäre, welches man dann um einen Kreis aus Nägeln spannt. Die vollen Farben befinden sich nun am Rand und der Unbuntpunkt (Weiß) in der Mitte am Koordinatenursprung. Das Koordinatensystem erhält die Bezeichnung U für die x- und V für die y-Achse.

Läßt man einen Zeiger vom Koordinatenursprung zu einer Farbe zeigen, gibt der Winkel des Zeigers den Farbton an. Ein Winkel von  $100^\circ$  entspricht dabei dem Farbton Rot,  $240^\circ$  etwa dem Farbton Grün und  $340^\circ$  etwa dem Farbton Blau.

Die Länge des Zeigers gibt die Farbsättigung der Farbe an (in der Videotechnik wird statt „Farbe“ der Begriff „Fartbart“ verwendet). Der Farbton wird also durch den Phasenwinkel und der Sättigungsgrad durch die Zeigerlänge ausgedrückt.

$$\text{Fartbart} = \text{Farbton} + \text{Farbsättigung}$$

### 7.7.2.2 Erzeugung des Y-Signals

Bei der Farbfernsehtechnik wird mit drei Röhren gearbeitet, welche die Spannungen der Komponenten Rot, Grün und Blau liefern. Diese Spannungen entsprechen der physikalischen Helligkeit der Bildpunkte, was nicht der Farbempfindlichkeit des menschlichen Auges für die drei Grundfarben entspricht. Das Y-Signal wird deshalb unter Berücksichtigung dieser mit folgender Formel ermittelt.

$$Y = 0.30 \cdot U_{\text{Rot}} + 0.59 \cdot U_{\text{Grün}} + 0.11 \cdot U_{\text{Blau}}$$

Dies entspricht dem physiologischen Helligkeitsempfinden des Menschen und damit auch dem Videosignal eines Schwarzweißempfängers.

### 7.7.2.3 Erzeugung des U- und V-Signals

Da man aus den drei Signalen R, G, B schon ein viertes Signal (Y) erzeugt hat, der auf jeden Fall übertragen wird, genügt es, nur noch zwei weitere Signale zu erzeugen und zu übertragen. Dann können alle drei Ausgangssignale (R, G, B) aus den übertragenen drei Signalen (Y, U, V) rekonstruiert werden. In der Praxis ignoriert man den Grünanteil (der steckt schon im Y-Signal drin) und überträgt statt dessen das V- und das U-Signal, welche wie folgt berechnet werden:

$$V = R - Y \quad U = B - Y \quad (R = U_{\text{Rot}} \quad B = U_{\text{Blau}}).$$

Diese beiden Signale enthalten nun zusammen mit dem Y-Signal die gesamte Farbinformation, zusammengemischt ergeben die beiden Signale das Chrominanzsignal.

### 7.7.3 Die Farbauflösung des PAL-Systems

Bei der Entwicklung farbiger Videosysteme stellte man fest, daß es genügt, nur das Helligkeitssignal (Schwarzweißbild) mit einer hohen Frequenz zu übertragen, während die Auflösung des Farbsignals wesentlich geringer sein kann. Die beiden Signale ergeben trotzdem bei der Überlagerung auf dem Bildschirm ein wirkungsvolles Farbbild. Der Effekt, der hier entsteht, läßt sich mit einer feinen Schwarzweißzeichnung vergleichen, die grob mit Buntstiften ausgemalt wird. Es ergibt sich trotz der geringen Farbauflösung ein detailreiches Farbbild.

Bei der Übertragung des PAL-Farbfernsehens wird die Bandbreite des Chrominanz-Signals auf 1.3 MHz begrenzt, so daß sich die Farbart wesentlich langsamer ändert als die Helligkeitsinformation. Die Farbauflösung von *VLab* liegt etwas über dem Farbinformationsgehalt des Fernsehbildes; daher tritt bei der Digitalisierung kein Informationsverlust auf.

### Literaturhinweis

Für den an der Fernsehtechnik interessierten Anwender empfiehlt sich das Buch:

Limann/Pelka: Fernsehtechnik ohne Ballast,  
erschienen im Franzis-Verlag, 16. Auflage 1991

# Kapitel 8

## Makros und ARexx

### 8.1 Allgemeines

In diesem Kapitel werden die verfügbaren Makro- bzw. ARexx-Kommandos der *VLab*-Software beschrieben. Das Programm wurde so konzipiert, daß alle interessanten Funktionen und Einstellungen dem Benutzer in Form von eigenen ARexx-Befehlen zugänglich sind. Die vorliegenden Befehlserklärungen gelten für die *VLab*-Version 4.0. Aktualisierte Fassungen findet man auf den jeweiligen Programm-Disketten.

Eine ausführliche Dokumentation der Prozeßsteuerungssprache ARexx kann man im „Handbuch zur Systemsoftware“ der Betriebssystemversion 2.X nachlesen. Wenn man ARexx einsetzen möchte, sollte man darauf achten, daß das Betriebssystem durch „as-signs“ usw. dafür richtig eingerichtet ist.

Der Anwender kann die ARexx-Kommandos von *VLab* auf unterschiedliche Art und Weise nutzen:

- Als einzelne ARexx-Befehle:  
Einzelne Operationen können mit `Befehl` ausführen ausgelöst werden oder mit Hilfe des mitgelieferten ARexx-Skripts „VShell.rx“ nacheinander eingetippt werden.
- Als *VLab*-Makros:  
Es können mit einem gewöhnlichen Texteditor ganze Befehlsfolgen mit *VLab*-ARexx-Befehlen zusammengestellt werden. Diese „Makros“ werden dann mit `Makro Laden` in das Benutzermenü eingebunden und von dort aus gestartet. Ein *VLab*-Makro muß mit der Kennung `#VLAB` beginnen und darf ausschließlich *VLab*-Kommandos enthalten. Im Vergleich zu normalen ARexx-Skripten sind *VLab*-Makros schneller und sparen Speicher, können jedoch nicht so vielseitig eingesetzt werden.

- Als ARexx-Skripts:

Natürlich kann man *VLab*-ARexx-Kommandos mit gewöhnlichen ARexx-Befehlen gemeinsam verwenden und so zu neuen ARexx-Skripts kombinieren. Diese lassen sich wie Makros vom Benutzermenü aus aufrufen, wobei (falls noch nicht geschehen) „RexxMast“ von *VLab* automatisch aktiviert wird. Sie könnten jedoch auch anders (etwa durch Doppelklick auf ein Piktogramm) gestartet werden.

Außerdem kann der Anwender vom Benutzermenü aus Amiga-Skript-Dateien ablaufen und sogar Programme starten lassen. *VLab* selbst benutzt übrigens auch ein Makro namens *VLab.Startup*, das beim Aufruf des Programms abgearbeitet wird. Dieses kann und soll vom Benutzer abgeändert oder erweitert werden.

## 8.2 Beispiel für ein Vlab-Makro

Es folgt ein Beispiel für ein *VLab*-Makro, das ein farbiges Vollbild aus der Quelle „Default“ einliest, die Delinterlace-Operation aufruft, eine Graustufen-Vorschau auf einem eigenem Screen durchführt, das Bild in HAM umrechnet und anzeigt:

```
#VLAB
;
SAVESETTINGS T:Kurzzeit           ; Einstellungen zwischenspeichern
SELECTSOURCE Default              ; Quelle "Default" anwählen
CLEAR ALL FORCE                    ; alle Puffer löschen
;
SHOWTITLE ON                      ; Titelleiste einschalten
CLOSEONRMB ON                    ; rechte Maustaste schließt Screen
AUTOPREVIEW OFF                  ; keine automatische Vorschau
AUTODISPLAY OFF                  ; keine automatische Anzeige
;
DITHER OFF                       ; Schattierung aus
SPECIALCONVERT OFF               ; keine Spezialumrechnungen
CONVERTDEPTH 6                   ; Farbtiefe=6 (für normales HAM)
COLORFIDELITY 254                ; volle Farbtreue
TRUECOLOR                        ; keine Farbkorrektur
;
SCANMODE LACE COLOR HIRES        ; Einlese-Modus setzen
PREVIEWMODE NOSCALE LACE BW      ; Vorschau-Modus setzen
PREVIEWOUTPUT SCREEN             ; Vorschau auf eigenem Screen
CONVERTMODE LORES HAM            ; Umrechnen nach HAM
;
```

```

SCAN                               ; Digitalisierung starten
DEINTERLACE 10                     ; DeInterlace ausführen
PREVIEW                             ; Vorschau zeigen
CONVERT                             ; Umrechnung durchführen
DISPLAY                             ; Amigabild darstellen
LOADSETTINGS T:Kurzzeit            ; Einstellungen wiederherstellen

```

### 8.3 Der CLI-Befehl VLabCmd

VLabCmd ist ein kurzes CLI-Kommando, das in der Lage ist, *VLab*-ARexx-Befehle an *VLab* zu senden und die Ergebnisse wieder zu empfangen. Viele Amiga-Anwender haben sich sicherlich noch nie intensiv mit der Programmiersprache ARexx beschäftigt und es fällt ihnen daher schwer, *VLab* über ein ARexx-Skript zu steuern.

Die Skriptsprache von Amiga-OS 2.X, wie sie z.B. in der *startup-sequence* benutzt wird, ist zwar nicht so leistungsfähig wie ARexx, dafür ist sie einfacher zu erlernen und fast alle Anwender sind damit weitgehend vertraut. Durch VLabCmd wird es möglich *VLab* recht komfortabel aus solchen Batch-Dateien heraus zu steuern (Beispiel siehe 8.4.2).

Das Template von VLabCmd lautet: *Command/A/F*. Das bedeutet, daß VLabCmd genau einen Parameter benötigt, der aus dem kompletten Rest der Zeile besteht. Wegen des /F-Parameters dürfen am Zeilenende keine Zeichen mehr folgen, die nicht zum Parameter gehören. Das Semikolon eines Kommentars muß daher direkt hinter dem Argument folgen:

```

Falsch:  VLabCmd SCREENTOFront    ;Screen nach vorne holen
Richtig: VLabCmd SCREENTOFront;   Screen nach vorne holen

```

VLabCmd unterstützt auch beide Rückgabewerte der *VLab*-Funktionen RC und RESULT, wie sie im Handbuch bei den einzelnen ARexx-Befehlen dokumentiert sind. RC wird in den normalen Return-Code umgesetzt und kann mit *If WARN*, *If NOT ERROR* usw. abgefragt werden. RESULT wird, wenn vorhanden, als Text an die Standard-Ausgabe (in der Regel das Shell-Fenster) ausgegeben. Er kann aber auch einer Umgebungsvariablen zugewiesen werden.

## 8.4 Mitgelieferte Makros

### 8.4.1 Edit.rx

Das ARexx-Skript `Edit.rx` ermöglicht es, auf einfache Art kleinere Änderungen an einem *VLab*-Makro oder ARexx-Skript vorzunehmen. Nach Aufruf des Makros wird ein File-Requester geöffnet. Darüber kann die zu editierende Datei ausgewählt werden. Sie wird in den Editor „Ed“ geladen, der sein Fenster auf dem *VLab*-Screen öffnet. Nach Durchführung der Änderungen kann die Datei mit „ESC x“ gespeichert werden, Ed wird danach beendet. Abschließend wird *VLab* noch über die Änderung der Datei informiert.

### 8.4.2 Edit.sc

Das Skript `Edit.sc` ist von der Funktion her identisch mit dem ARexx-Skript `Edit.rx` (siehe dort). Es ist allerdings nicht in ARexx geschrieben, sondern einfach als Skript, wie z.B. auch die `startup-sequence`. `Edit.sc` ist ein Beispiel für den Einsatz des Befehls `VLabCmd`.

### 8.4.3 ParseSequence.rx

Bei den Bildern einer Sequenz befinden sich in der Regel viele, die man wieder löschen möchte. Das wird durch diese Makro erleichtert. Nach Aufruf des Makros wird ein File-Requester geöffnet. Darüber kann die zu bearbeitende Sequenz gewählt werden. Danach werden alle Bilder der Sequenz der Reihe nach geladen und angezeigt. Über einen Requester kann dann gewählt werden, ob das Bild gelöscht werden soll oder nicht, oder ob das Skript abgebrochen werden soll.

### 8.4.4 ScanTestLines.vm

In der vertikalen Austastlücke gibt es bei Fernsehübertragungen einige Zeilen, die eine Reihe von Testsignalen enthalten und damit das bekannte Testbild praktisch ersetzen. Der große Vorteil ist, daß diese Zeilen ständig verfügbar sind, wohingegen das Testbild nur noch selten ausgestrahlt wird.

`ScanTestLines.vm` ist ein *VLab*-Makro und arbeitet daher auch ohne ARexx. Wird es gestartet, werden die oben beschriebenen Zeilen eingelesen und über `VZOOM` vergrößert, damit man sie später einfacher betrachten kann.

Mit Hilfe dieser Testzeilen läßt sich gut die Bildqualität des empfangenen Senders auf Rauschen oder Geisterbilder untersuchen. Auch die Auswirkung der speziellen *VLab*-Filter (Luminanz, Chrominanz, Filterfrequenz und Rauschfilter) läßt sich gut beobachten. An einem normalen, sich ständig verändernden Bild fehlt in der Regel eine Vergleichsmöglichkeit.

#### **8.4.5 SequenceToAmiga.rx**

Dieses ARexx-Skript konvertiert eine vollständige *VLab*-Sequenz in Amiga-IFF-Files, so daß diese Bilder dann weiterverarbeitet werden können. Nach Aufruf des Makros wird ein File-Requester geöffnet. Darüber kann die zu konvertierende Sequenz ausgewählt werden. Danach werden alle Bilder der Sequenz der Reihe nach geladen, umgerechnet und wieder gespeichert. Über einen Requester kann am Ende entschieden werden, ob die Bilder als Film angezeigt werden sollen oder nicht. Die Darstellung erfolgt über das Utility „display“, welches mit „Ctrl-D“ wieder verlassen werden kann.

Die Art der Umrechnung richtet sich automatisch nach dem YUV-Typ. Farbbilder werden in HAM und Schwarzweißbilder in Schwarzweiß (Hires oder Lores) umgerechnet. Änderungen an diesem Verhalten (z.B. Floyd-Steinberg) können leicht durch Änderung des Skripts erreicht werden.

Es ist zu beachten, daß die YUV-Bilder von den berechneten Amiga-Bildern überschrieben werden, die YUV-Bilder gehen also verloren. Auch dieses Verhalten kann der Anwender leicht selbst ändern.

#### **8.4.6 SetSpecialValues.rx**

*VLab* kennt einige Einstellungen, die nicht über das Menü oder ein Gadget verändert werden können. Lediglich über einen ARexx-Befehl lassen sich diese Einstellungen variieren. Das ARexx-Skript *SetSpecialValues.rx* ermöglicht eine einfache Kontrolle oder Abänderung der wichtigsten Werte.

Nacheinander wird für jeden Befehl ein Requester geöffnet. Darin wird der Befehl (Command:) und dessen aktuelle Einstellung (Value:) angezeigt. Durch Anklicken des Gadgets „OK“ wird dieser Wert unverändert belassen. „Cancel“ bricht das Skript komplett ab. Die Gadgets zwischen „OK“ und „Cancel“ stellen alle möglichen Werte für den aktuellen Befehl dar. Durch Anklicken des gewünschten Gadgets wird die Einstellung für den Befehl entsprechend geändert.

### 8.4.7 VLABtoYUVN.rx

Aus Geschwindigkeitsgründen werden die Bilder beim Einlesen einer Sequenz nicht im normalen YUVN-Format, sondern in einem internen *VLab*-Format gespeichert. *VLab* selbst kann natürlich beide Formate lesen, anderen Programmen ist das interne Format dagegen unbekannt.

*VLABtoYUVN.rx* konvertiert daher eine komplette Sequenz vom *VLab*- in das YUVN-Format. Nach Aufruf des Makros wird ein File-Requester geöffnet. Darüber kann die zu konvertierende Sequenz ausgesucht werden. Danach kann noch über einen Requester bestimmt werden, ob bei jedem einzelnen Bild ein „DeInterlace“ durchgeführt werden soll.

### 8.4.8 VShell.rx

*VLab* kann außer über seine Menüs und Gadgets in verschiedenen Fenstern auch über *ARexx* gesteuert werden. Manchmal ist es recht praktisch, die Befehle mit Hilfe einer Shell einzutippen, das ist natürlich auch vom Geschmack des Benutzers abhängig. Das *ARexx*-Skript *VShell.rx* bietet diese Möglichkeit.

Wird *VShell.rx* gestartet, wird ein Fenster auf dem *VLab*-Screen geöffnet. Darin kann man, wie in der normalen Amiga-Shell, Befehle mit ihren Parametern eingeben, allerdings nur solche Befehle, die *VLab* versteht. *ARexx*- oder CLI-Befehle lassen sich hingegen nicht aufrufen.

Die *VLab*-Shell wird durch Anklicken des Close-Gadgets beendet. Solange sie nicht beendet und damit das Fenster geschlossen ist, kann *VLab* seinen Screen nicht schließen.

### 8.4.9 LoMem.vm und HiMem.vm

Das *VLab*-Makro *LoMem.vm* sollte aufgerufen werden, wenn im System nur wenig RAM (etwa 1MB) vorhanden ist. Es stellt verschiedene Funktionen so um, daß möglichst wenig Speicher benötigt wird. Natürlich wird dadurch die Leistungsfähigkeit von *VLab* beschränkt. *HiMem.vm* stellt den Normalbetrieb der Operationen (soweit möglich) wieder her.

## 8.5 Kurzbeschreibung der Parameterarten

In der Befehlliste wird unter „Schablone“ erklärt, welche Optionen verwendet werden müssen oder können. Dabei wird folgende Symbolik benutzt:



- kein / Der Parameter kann, muß aber nicht angegeben werden.
- /A Der Parameter muß angegeben werden.
- /N Es kann eine Zahl angegeben werden.
- /A/N Es muß eine Zahl angegeben werden.
- /S Der Parameter ist ein Schlüsselwort als Schalter.
- /M Es können ein oder mehrere Parameter angegeben werden.
- /K Der Parameter muß (wenn er angegeben wird) seinem Schlüsselwort folgen.

Folgt dem Kommando beim Aufruf statt einer Parameterliste ein Fragezeichen „?“, wird die zugehörige Parameterschablone aufgelistet.

Es ist zu beachten, daß negative Zahlen in ARexx-Skripts von Anführungsstrichen umrahmt werden müssen.

## 8.6 Rückgabewerte und Fehlercodes

Es folgen die Rückgabewerte und Fehlercodes beim Aufruf der Kommandos über ARexx.

Es existieren folgende Fehlercodes:

RC	=	OK	=	0:	alles in Ordnung
		WARN	=	5:	Warnung, Abbruch im File-Requester etc.
		ERROR	=	10:	Kommando konnte nicht richtig ausgeführt werden
		FATAL	=	20:	Fehlerhafte, bzw. fehlende Parameter
		FAIL	=	-1:	Kein VLab-Kommando

Schließen sich /S-Parameter gegenseitig aus, oder muß mindestens einer angegeben werden, erhält man einen Fehlercode von 20 (FATAL). Beispiel:

AUTOUSEENV	⇒	RC=FATAL
AUTOUSEENV ON OFF	⇒	RC=FATAL
AUTOUSEENV ON	⇒	RC=OK
AUTOUSEENV OFF	⇒	RC=OK

Rückgaben erfolgen über die ARexx-Variable RESULT. Dazu muß der zugehörige Fehlercode 0 sein (RC=OK). Die jeweils möglichen Rückgaben sind bei den Kommandos angegeben. Außerdem können über das Kommando READ bei den mit ⊗ gekennzeichneten Kommandos die aktuellen Einstellungen ausgelesen werden.

## 8.7 Die einzelnen ARexx-Befehle

Kommando: ABOUT

Schablone: Keine Parameter

Zweck: Es erscheint ein Requester mit einigen Informationen zum Programm. Zusätzlich zu den üblichen Mitteilungen, wie Versionsnummer und Hersteller, wird auch der Name des ARexx Ports und des Public Screens von *VLab* ausgegeben.

Wird *VLab* mehrmals gestartet, hat nur das zuerst gestartete Programm einen ARexx-Port. Diese Angabe entfällt dann bei den anderen.

---

Kommando: ACTIVATEWINDOW

Schablone: SCAN/S, CONVERT/S, COLOR/S, SELECTSOURCE/S,  
DEFINESOURCE/S, PREVIEW/S, SEQUENCE/S, IFR/S,  
MONITOR/S, COMMAND/S

Zweck: Mit diesem Kommando kann ein *VLab*-Fenster aktiviert werden. Werden mehrere Fenster angegeben, wird nur das erste Schlüsselwort in der Schablone verwendet. Durch Aufruf ohne Parameter aktiviert man das Hauptfenster. Ist das zu aktivierende Fenster nicht geöffnet, wird ein Fehlercode zurückgegeben.

Beispiele: ACTIVATEWINDOW SCAN

Das Fenster, welches zur Steuerung des Einlesevorganges dient wird aktiviert.

ACTIVATEWINDOW

Das Hauptfenster wird aktiviert.

RC: ERROR, wenn Fenster nicht offen

---

Kommando: ADDSOURCE

Schablone: FORCE/S

Zweck: Die durch die Befehle SOURCENAME, INPUT, PALMODE, etc. getätigten Einstellungen der Hardware werden mit diesem Befehl für eine spätere Verwendung gesichert.

Beispiele: ADDSOURCE FORCE

Sollte bereits eine Definition mit gleichem Namen existieren, wird diese ohne Sicherheitsabfrage überschrieben.

RC: WARN, bei Abbruch durch Sicherheitsabfrage

---

Kommando: AREXXDEBUG ⊗

Schablone: ON/S,OFF/S

Zweck: Über dieses Kommando kann ein spezieller Debug-Modus für AREXX-Skripts ein- bzw. ausgeschaltet werden.

Beispiele: AREXXDEBUG ON

Alle folgenden AREXX-Kommandos, sowie ihre Rückgabewerte, werden in einem Fenster ausgegeben.

AREXXDEBUG OFF

Beenden des Debug-Modus.

---

Kommando: ASLREQUEST

Schablone: TITLE/A,FILE/K,SAVE/S,DIRS/S

Zweck: Über dieses Kommando kann ein ASL-Requester geöffnet werden. Der Text der Titelleiste muß angegeben werden. Pfad und Pattern müssen bei Bedarf vorher über SETPATH USER bzw. SETPATTERN USER angegeben werden. SAVE wählt den Requester zum Speichern. Ist DIRS angegeben, werden nur Verzeichnisse und keine Dateien angezeigt.

Beispiele: ASLREQUEST "Datei speichern:" FILE "Testdatei" SAVE

ASLREQUEST "Verzeichnis wählen:" DIRS

RC: WARN, wenn Requester mit „Cancel“ abgebrochen wurde oder kein Filename angegeben wurde.

RESULT: Der vollständige Dateiname mit Pfad

---

Kommando: AUTODISPLAY ⊗

Schablone: ON/S,OFF/S

Zweck: Wenn aktiviert, wird das Bild nach der Konvertierung in ein Amiga-Bild automatisch angezeigt.

---

Kommando: AUTOPREVIEW ⊗

Schablone: ON/S,OFF/S

Zweck: Falls aktiv, wird immer, wenn sich das YUV-Bild im Speicher geändert hat, (z.B.: Einlesen, Laden) automatisch eine neue Vorschau des Bildes berechnet und angezeigt.

---

Kommando: AUTOSAVEENV ⊗

Schablone: ON/S,OFF/S

Zweck: Entscheidung, ob die Environment-Variable, welche die Einstellungen von *VLab* enthält, automatisch beim Beenden von *VLab* gespeichert werden soll. Wird AUTOSAVEENV eingestellt, so wird automatisch auch AUTOUSEENV aktiviert.

Beispiele: AUTOSAVEENV ON

Die Environment-Variable soll automatisch gespeichert werden. *VLab* startet dadurch selbst nach einem Reset in der Konfiguration, die beim letzten Verlassen vorlag.

AUTOSAVEENV OFF

Die Umgebungsvariable soll nicht automatisch gespeichert werden.

---

Kommando: AUTOSCROLL ⊗

Schablone: ON/S,OFF/S

Zweck: Arbeitet *VLab* auf einem eigenen Screen und ist dieser größer als die Workbench, kann hiermit bestimmt werden, ob der Programm-Screen von *VLab* automatisch scrollen soll, wenn der Mauszeiger den Bildschirmrand erreicht, oder nicht.

---

Kommando: AUTOUNDO ⊗

Schablone: ON/S,OFF/S

Zweck: Wenn aktiv, wird vor jeder Änderung des Bildes im YUV-Puffer (z.B.: Einlesen, Laden), automatisch eine Kopie des Bildes angelegt. Über das Kommando UNDO kann die jeweils letzte Änderung rückgängig gemacht werden.

---

Kommando: AUTOUSEENV ⊗

Schablone: ON/S,OFF/S

Zweck: Umschaltung, ob die temporäre Environment-Variable, welche die Einstellungen von *VLab* enthält, automatisch beim Beenden von *VLab* aktualisiert werden soll. Wenn ja, startet *VLab* bis zum nächsten Reset immer in der Konfiguration, die beim letzten Verlassen geherrscht hat.

Beispiele: AUTOUSEENV OFF  
Die Environment-Variable soll nicht automatisch aktualisiert werden.

---

Kommando: BINDVLABPAR

Schablone: Keine Parameter

Zweck: Falls die *VLab*-Software gestartet wurde, ohne vorher die Hardware einzuschalten (nur bei Parallelport-Version), kann der Software hiermit mitgeteilt werden, daß die Hardware jetzt betriebsbereit ist.

---

Kommando: BLACKBACKGROUND ⊗

Schablone: ON/S,OFF/S

Zweck: Bei der Konvertierung eines YUV- in ein Amiga-Farbbild kann entschieden werden, ob auf jeden Fall ein schwarzer Hintergrund erzeugt werden soll oder nicht.

---

Kommando: BLITMODE ⊗

Schablone: AUTO/S,CLIPBLIT/S,BLITTER/S,CPU/S

Zweck: Wahl der Ausgabemethode für das Monitorfenster.

Beispiele: BLITMODE AUTO  
*VLab* wählt automatisch den optimalen Modus, der höchste Geschwindigkeit mit möglichst wenig Flackern bedeutet. In der Regel liefert BLITMODE AUTO die besten Ergebnisse.

#### BLITMODE CLIPBLIT

Dieser Modus wird immer dann benutzt (auch bei AUTO), wenn das Monitorfenster teilweise von einem anderen Fenster verdeckt wird. Wurde speziell BLITMODE CLIPBLIT gewählt, wird der Modus immer verwendet; die Automatik ist dann abgeschaltet.

#### BLITMODE BLITTER

Wurde „BLITTER“ eingestellt, wird immer eine spezielle, zeilenorientierte Blitter-Routine benutzt, die auch bei acht oder 16 Graustufen sehr wenig flackert.

#### BLITMODE CPU

Hat der benutzte Amiga eine sehr schnelle CPU (68030/40) kann die Darstellungsgeschwindigkeit über diesen Modus evtl. noch weiter erhöht werden. Zu beachten ist, daß das Monitorfenster dazu an bestimmten Positionen stehen muß, *VLab* verschiebt das Fenster daher bei Bedarf.

---

Kommando: BREAK

Schablone: ALL/S,C/S,D/S,E/S,F/S

Zweck: Beim nächsten eingehenden ARexx-Kommando werden bei dem sendenden Task die angegebenen Signalbits gesetzt. Beim Neustart von ARexx-Skripts von *VLab* aus wird die Break-Aufforderung vorher gelöscht.

---

Kommando: CHANGEASLREQ

Schablone: PRESETS/S,MACROS/S,YUV/S,24BIT/S,AMIGA/S,  
DELETE/S,SEQUENCE/S,IFR/S,REWIND/S,  
CONVERT/S,USER/S,FONT/S,  
LEFTEDGE/N,TOPEDGE/N,WIDTH/N,HEIGHT/N

Zweck: Festlegung der Position und Größe der ASL-Requester. Soll ein Wert nicht verändert werden, so ist -1 anzugeben.

Beispiele: CHANGEASLREQ YUV "-1" 120

Die Y-Position wird auf 120 gesetzt. X-Position, Breite und Höhe bleiben unverändert.

---

Kommando: CHANGEHELPWINDOW

Schablone: LEFTEDGE/N, TOPEDGE/N, WIDTH/N, HEIGHT/N, ALT/S

Zweck: Festlegung der Position und Größe des Help-Fensters. Soll ein Wert nicht verändert werden, so ist -1 anzugeben. In der momentanen Programmversion wird das Help-Fenster beim Aufruf dieses Kommandos geschlossen.

Beispiele: CHANGEHELPWINDOW "-1" 150 "-1" 400  
Die Y-Position wird auf 150 und die Breite auf 200 gesetzt. X-Position und Höhe bleiben unverändert.

CHANGEHELPWINDOW 50 200 640 200 ALT  
Die Alternativgröße und -position wird auf die angegebenen Werte gesetzt.

---

Kommando: CHANGEPREVIEWWINDOW

Schablone: LEFTEDGE/N, TOPEDGE/N, WIDTH/N, HEIGHT/N

Zweck: Festlegung der Position und Größe des Vorschau-Fensters. Soll ein Wert nicht verändert werden, so ist -1 anzugeben.

Beispiele: CHANGEPREVIEWWINDOW "-1" "-1" 200  
Nur die Höhe wird verändert (auf 200).

---

Kommando: CHROMAFILTER ⊗

Schablone: ON/S, OFF/S

Zweck: Mittels dieses Kommandos wird der Chrominanzfilter auf der VLab-Hardware ein- oder ausgeschaltet.

Beispiele: CHROMAFILTER ON  
Der Farbfilter wird eingeschaltet.

---

Kommando: CLEAR

Schablone: YUV/S, UNDO/S, AMIGA/S, ALL/S, FORCE/S

Zweck: Löschen von Bildpuffern. Der Parameter FORCE unterdrückt ggf. Sicherheitsabfragen.

Beispiele: CLEAR AMIGA FORCE  
Der Puffer für die fertig berechneten Amiga-Bilder wird ohne Sicherheitsabfrage gelöscht.

CLEAR ALL

Alle internen Bildpuffer von *VLab* werden gelöscht.

RC: WARN, wenn bei der Sicherheitsabfrage abgebrochen wurde

---

Kommando: CLOSEONRMB ⊗

Schablone: ON/S,OFF/S

Zweck: Festlegung, ob die Screens für Vorschau oder Anzeige der fertig berechneten Amiga-Bilder durch Drücken der rechten Maustaste geschlossen werden können oder nicht.

Beispiele: CLOSEONRMB ON

Der Screen kann durch Druck auf die rechte Maustaste geschlossen werden.

---

Kommando: CLOSEPREVIEWSCREEN

Schablone: Keine Parameter

Zweck: Schließen des Screens für die Vorschau oder die Anzeige der fertig berechneten Amiga-Bilder.

---

Kommando: CLOSEWINDOW

Schablone: SCAN/S, CONVERT/S, COLOR/S, SELECTSOURCE/S,  
DEFINESOURCE/S, PREVIEW/S, SEQUENCE/S, IFR/S,  
MONITOR/S, COMMAND/S, HELP/S, ALL/S

Zweck: Schließen von *VLab*-eigenen Fenstern.

Beispiele: CLOSEWINDOW PREVIEW

Das Vorschau-Fenster wird geschlossen.

CLOSEWINDOW

Das gerade aktive Fenster wird geschlossen.

CLOSEWINDOW ALL

Alle *VLab*-Fenster werden geschlossen.

---

Kommando: COLORFIDELITY ⊗

Schablone: NUMBER/A/N



**Zweck:** Bestimmung der Farbtreue für die Umrechnung in HAM-Bilder. Ein Wert von 0 sorgt für eine möglichst realistische Reproduktion der Helligkeitswerte, bei 254 wird auf optimale Farben geachtet. Werte zwischen 0 und 254 sind möglich.

**RC:** FATAL, wenn Wert nicht im Bereich 0...254

---

**Kommando:** COLORVALUE

**Schablone:** R=RED/K/N,G=GREEN/K/N,B=BLUE/K/N,  
CHR=CHROMINANCE/K/N, LUM=LUMINANCE/K/N,  
CON=CONTRAST/K/N,GAM=GAMMA/K/N

**Zweck:** Über dieses Kommando kann eine Farbkorrektur eingestellt werden. Diese wird bei der Bildumrechnung sowie bei der Darstellung auf einer 24 Bit-Grafikkarte und beim 24 Bit-Export benutzt.

**Beispiele:** COLORVALUE R 50 B 110

Der Rotanteil soll halbiert und der Blauanteil etwas angehoben werden.

COLORVALUE CON 100

100 ist die neutrale Stellung (außer bei GAMMA und LUMINANCE) und bedeutet damit in diesem Beispiel, daß der Kontrast bei der Konvertierung nicht verändert werden soll.

**RC:** FATAL, wenn Wert nicht im Bereich von 0 bis 200 oder Wert nicht im Bereich von -100 bis 100 bei GAMMA oder LUMINANCE

---

**Kommando:** COMMANDERRORS



**Schablone:** ON/S,OFF/S

**Zweck:** Entscheidung, ob fehlerhafte Kommandos zu einem Fehlerrequester führen oder nicht.

**Beispiele:** COMMANDERRORS OFF

Bei fehlerhaften Kommandos erscheint kein Requester. Statt dessen wird in RESULT ein Fehlertext zurückgegeben.

---

**Kommando:** CONVERT

**Schablone:** Keine Parameter

**Zweck:** Das YUV-Bild wird in ein Amiga-IFF-Bild vom zuvor eingestellten Typ umgerechnet.

RC: WARN, wenn kein YUV-Bild vorliegt  
· WARN, wenn auf Grund der Sicherheitsabfrage abgebrochen wurde  
ERROR, wenn Konvertierung nicht funktioniert hat (Speichermangel,  
SW bei Farbkonvertierung, Abbruch wegen COLORFIDELITY, ...)

---

Kommando: CONVERTDEPTH ⊗

Schablone: DEPTH/A/N

Zweck: Mit diesem Kommando wird die Farbtiefe festgelegt, die das fertig konvertierte Amiga-Bild haben soll. Die Festlegung "LoRes" oder "HiRes" sollte durch das Kommando CONVERTMODE vorher erfolgen.

Beispiele: CONVERTDEPTH 3

Das Bild soll nach dem Umrechnen drei Bitplanes haben.

RC: ERROR, wenn die gewählte Tiefe sich nicht mit anderen Einstellungen verträgt (z.B. DEPTH=3 beim HAM-Modus)  
FATAL, bei DEPTH > 8

---

Kommando: CONVERTMODE ⊗

Schablone: LORES/S,HIRES/S,BW/S,COLOR/S,HAM/S

Zweck: Hiermit kann der Bildtyp gewählt werden, in den das YUV-Bild beim nächsten Aufruf von CONVERT umgerechnet wird. Da beim HAM-Modus die Anzahl der Bitplanes festgelegt ist, kann das Kommando CONVERTDEPTH bei diesem entfallen.

Beispiele: CONVERTMODE LORES HAM

Beim nächsten Bildkonvertieren erhält man ein LoRes-HAM-Bild. Die Anzahl der "Bitplanes" sollte mit CONVERTDEPTH auf „sechs“ gestellt werden.

CONVERTMODE HIRES HAM

Die Betriebsart „HiRes-HAM“ kann nur auf dem Amiga 4000 wiedergegeben werden. Die Farbtiefe kann auf „acht“ gesetzt werden, um die möglichen 262144 Farben nutzen zu können.

CONVERTMODE HIRES BW

Die nächste Bildkonvertierung liefert ein Schwarzweiß-HiRes-Bild. Hierfür sollte dann mit dem Kommando CONVERTDEPTH auch die Anzahl der Bitplanes festgelegt werden.

RC: ERROR, wenn ein Modus nicht möglich ist (z.B. HAM mit Floyd Steinberg).

---

Kommando: COPYTOUNDO

Schablone: Keine Parameter

Zweck: Das aktuelle YUV-Bild wird zusätzlich in einen zweiten Puffer kopiert, von wo es mit dem Kommando UNDO zurückgeholt werden kann.

RC: WARN, wenn kein YUV-Bild da ist  
ERROR, wenn nicht genug Speicher

---

Kommando: DEFAULTPUBSCREEN



Schablone: ON/S,OFF/S

Zweck: Festlegung, ob der Screen von *VLab* zum Default-Public-Screen gemacht werden soll oder nicht.

Beispiele: DEFAULTPUBSCREEN ON

Der *VLab*-Screen wird zum Default-Public-Screen gemacht, d.h., alle von nun an gestarteten Programme öffnen ihre Fenster auf dem *VLab*-Screen, sofern die Programme nicht einen eigenen Screen benutzen oder explizit den Workbench-Screen anfordern.

---

Kommando: DEFINECLIP

Schablone: X1/N,Y1/N,X2/N,Y2/N

Zweck: Wahl eines Bildausschnittes für den Digitalisierungsvorgang: Die angegebenen Koordinaten bestimmen den linken oberen und den rechten unteren Eckpunkt des gewünschten Bildbereiches.

Beispiele: DEFINECLIP 10 40 700 600

---

Kommando: DEINTERLACE

Schablone: LEVEL/N,EVEN/S,ODD/S,MIX/S

Zweck: Durch schnelle Bewegungen entstandene Verschiebungen in einem Vollbild werden weggerechnet. Dadurch wird starkes Flackern des Bildes verhindert, ohne auf die höhere Vollbild-Auflösung zu verzichten.

RC: FATAL, wenn  $LEVEL \leq 0$  oder  $LEVEL > 255$   
WARN, wenn kein YUV-Bild im Speicher ist  
OK, wenn es kein Vollbild ist, bzw. die Funktion ausgeführt wurde

---

Kommando: DELETEFILE

Schablone: FILENAME,FORCE/S

Zweck: Löschen einer Datei mit ihrem eventuell vorhandenen Piktogramm.

Beispiele: DELETEFILE Bild FORCE

Die Datei mit dem Namen "Bild" und (falls vorhanden) auch noch "Bild.info" wird ohne Sicherheitsabfrage gelöscht, weil der Parameter FORCE angegeben ist.

DELETEFILE

Nach Öffnen eines File-Requesters kann eine Datei mit ihrem eventuell existierenden Icon gelöscht werden.

RC: WARN, wenn der ASL-Requester abgebrochen wurde  
ERROR, wenn Delete nicht funktionierte

---

Kommando: DELETEMACRO

Schablone: NAME/M,ALL/S,FORCE/S

Zweck: Mit diesem Kommando können Makros aus dem Speicher entfernt werden. Der Parameter FORCE unterbindet mögliche Sicherheitsabfragen. Wird der Parameter ALL angegeben, werden alle Makros aus dem Speicher entfernt. Die Dateien der Makros auf Festplatte oder Diskette werden hingegen nicht gelöscht.

Beispiele: DELETEMACRO VShell.rx

Das Makro mit dem Namen „VShell.rx“ wird aus dem Speicher entfernt.

RC: WARN, eine Sicherheitsabfrage wurde nicht bestätigt.  
ERROR, wenn ein Makro nicht aus dem Speicher entfernt werden konnte, weil es etwa nicht vorhanden war, oder gerade abgearbeitet wird.

---

Kommando: DELETESOURCE

Schablone: NAME/M,ALL/S,FORCE/S

**Zweck:** Mit diesem Kommando können Quelldefinitionen gelöscht werden. Der Parameter FORCE unterbindet mögliche Sicherheitsabfragen. Wird der Parameter ALL angegeben, werden alle Quelldefinitionen gelöscht.

**Beispiele:** DELETESOURCE Kamera

Die Definition mit dem Namen Kamera wird gelöscht.

DELETESOURCE ARD RTL Recorder

Die drei angegebenen Definitionen werden gelöscht.

DELETESOURCE

Wird DELETESOURCE ohne Parameter aufgerufen, wird das Definitionsfenster geöffnet.

**RC:** WARN, wenn eine Sicherheitsabfrage nicht bestätigt wurde.

ERROR, wenn eine oder mehrere Definitionen nicht gelöscht werden konnten, da sie z.B. nicht vorhanden waren.

---

**Kommando:** DISPLAY

**Schablone:** Keine Parameter

**Zweck:** Ein zuvor berechnetes Amiga-Bild wird angezeigt.

**RC:** WARN, wenn kein Amiga-Bild existiert

ERROR, Bild konnte nicht angezeigt werden (z.B. kein Amiga-Format)

---

**Kommando:** DISPLAYID



**Schablone:** ID/A/N

**Zweck:** Dieser Befehl dient der Änderung der Display-ID des aktuellen Amiga-Bildes. Die Definitionen der Display-IDs sind den Amiga-OS 2.X Include-Files zu entnehmen.

**Beispiele:** DISPLAYID 0

Das Amiga-Bild wird LoRes-Non-Interlace.

DISPLAYID 4

Das Amiga-Bild wird LoRes-Interlace.

DISPLAYID 32768

Das Amiga-Bild wird HiRes-Non-Interlace.

DISPLAYID 2052

Das Amiga-Bild wird Ham-Interlace.

RC: ERROR, wenn kein Amiga-Bild vorhanden ist.

---

Kommando: DISPLAYOUTPUT ⊗

Schablone: AMIGA/S,RETINA/S

Zweck: Mit dieser Option kann die Darstellung der in Amiga-Grafiken umgewandelten Videobilder auf die *Retina*-Grafikarte umgelenkt werden. Es können dabei sämtliche Amiga-Bildschirmmodi angezeigt werden.

---

Kommando: DITHER ⊗

Schablone: ON/S,OFF/S

Zweck: Wenn aktiv, wird bei der Umrechnung eines YUV- in ein Amiga-IFF-Bild der Floyd-Steinberg Algorithmus eingesetzt. Durch "Punktieren" des Bildes wird auch bei wenig Graustufen oder Farben ein gutes Ergebnis erzielt.

Beispiele: DITHER ON

Bei der nächsten Umrechnung wird der Floyd-Steinberg Algorithmus eingesetzt.

---

Kommando: ERASEPREVIEW

Schablone: Keine Parameter

Zweck: Dieses Kommando löscht das Vorschau-Fenster.

RC: ERROR, wenn das Fenster bzw. der Screen nicht offen war.

---

Kommando: EXPORT24

Schablone: FILENAME,DEEP/S,FORCE/S

Zweck: Das aktuelle YUV-Bild wird in ein RGB-24 Bit-Bild konvertiert und als IFF-Datei gespeichert. Beim Aufruf ohne Dateinamen wird der File-Requester geöffnet. Zum Speichern von YUV-Bildern sollte dieses Kommando nicht benutzt werden, da die erzeugte Datei sehr lang wird.

Beispiele: EXPORT24 DEEP

Das aktuelle YUV-Bild wird in ein RGB-24 Bit-Bild gewandelt und als IFF-DEEP-Datei gespeichert. Dieses recht neue Format kann u.U. noch nicht von allen Programmen zur Bildverarbeitung auf dem Amiga gelesen werden.

EXPORT24

Das aktuelle YUV-Bild wird in ein RGB-24 Bit-Bild konvertiert und als IFF-ILBM-Datei gespeichert.

RC: WARN, wenn kein YUV-Bild da ist  
WARN, wenn der ASL-Requester abgebrochen wurde  
WARN, wenn abgebrochen wurde, weil die Datei schon existiert  
ERROR, wenn SAVE nicht erfolgreich war.

---

Kommando: EXPORTAMIGA

Schablone: FILENAME,FORCE/S

Zweck: Ein zuvor berechnetes Amiga-IFF-Bild wird im IFF-Format gespeichert. Beim Aufruf ohne Parameter wird ein File-Requester geöffnet.

RC: WARN, wenn kein Amiga-Bild da ist  
WARN, wenn der ASL-Requester abgebrochen wurde  
WARN, wenn abgebrochen wurde, weil die Datei schon existiert  
ERROR, wenn SAVE nicht funktionierte

---

Kommando: FILTERFREQUENCY



Schablone: 2.5/S,3/S,3.5/S,4/S

Zweck: Wahl der Frequenz des Bandpaßfilters zur Veränderung der Bildschärfe. Wird nur benutzt, wenn FILTERWEIGHT nicht OFF ist.

---

Kommando: FILTERWEIGHT



Schablone: OFF/S,LIGHT/S,MEDIUM/S,STRONG/S

Zweck: Hiermit wird eingestellt, wie stark sich FILTERFREQUENCY und NOISEFILTER auf das zu digitalisierende Bild auswirken. In Stellung OFF sind beide Funktionen ohne Wirkung.

---

Kommando: FONT



Schablone: NAME,SIZE/N,DEFAULT/S,USER/S

Zweck: Wahl eines Zeichensatzes für die VLab-Fenster.

Beispiele: FONT Topaz 8

Es wird der Zeichensatz „Topaz 8“ gewählt.

FONT

Ein Fontrequester erscheint, mit dem die Auswahl vorgenommen wird.

FONT USER

Der zuletzt über den Fontrequester gewählte Zeichensatz wird genommen.

FONT DEFAULT

Der System-Default-Text-Font wird gewählt.

RC: WARN, bei Abbruch im Fontrequester

---

Kommando: GETCOLORVALUE

Schablone: R=RED/S,G=GREEN/S,B=BLUE/S,  
CHR=CHROMINANCE/S,LUM=LUMINANCE/S,  
CON=CONTRAST/S,GAM=GAMMA/S

Zweck: Diese Funktion gibt die eingestellten Werte der Farbkonvertierung zurück.

RESULT: Zahl zwischen 0 und 200 bzw. -100 und 100 bei GAMMA und LUMINANCE

---

Kommando: GETFILENAME

Schablone: PRESETS/S,MACROS/S,YUV/S,24BIT/S,AMIGA/S,  
DELETE/S,SEQUENCE/S,IFR/S,REWIND/S,  
CONVERT/S,USER/S

Zweck: Der Dateiname aus einem der angegebenen File-Requester kann ausgelesen werden.

RESULT: Dateiname

---

Kommando: GETFULLNAME



Schablone: PRESETS/S,MACROS/S,YUV/S,24BIT/S,AMIGA/S,  
DELETE/S,SEQUENCE/S,IFR/S,REWIND/S,  
CONVERT/S,USER/S

Zweck: Der Dateiname mit zugehörigen Pfad aus einem der angegebenen File-Requester kann ausgelesen werden.

RESULT: Dateiname mit vollständigem Pfad

---

Kommando: GETPATH

Schablone: PRESETS/S,MACROS/S,YUV/S,24BIT/S,AMIGA/S,  
DELETE/S,SEQUENCE/S,IFR/S,REWIND/S,  
CONVERT/S,USER/S

Zweck: Der Pfad aus einem der angegebenen File-Requester kann ausgelesen werden.

RESULT: Pfad

---

Kommando: GETPATTERN

Schablone: PRESETS/S,MACROS/S,YUV/S,24BIT/S,AMIGA/S,  
DELETE/S,SEQUENCE/S,USER/S

Zweck: Der Pattern-String aus einem der angegebenen File-Requester kann ausgelesen werden.

RESULT: Pattern

---

Kommando: GETPUBSCREENNAME

Schablone: Keine Parameter

Zweck: Ermittlung des Public-Screen-Namens von VLab. Ist kein eigener Screen geöffnet, gibt dieses Kommando den Namen "Workbench" zurück, sonst "VLAB.1" bzw. "VLAB.2" etc. In ARexx-Skripts sollten keine konstanten Namen, sondern immer die Resultate dieser Funktion verwendet werden.

RESULT: Name des Public-Screens

---

Kommando: GETSOURCENUMBER

Schablone: Keine Parameter

Zweck: Über dieses Kommando kann die Anzahl der zur Verfügung stehenden Videoeingänge ermittelt werden. Bei einer Karte sind dies zwei. Ist mehr als eine Platine im System, erhält man entsprechend mehr Videoeingänge.

RESULT: Anzahl der Videoeingänge, 0 = Keine Hardware gefunden

---

Kommando: GETVISITORNUMBER

Schablone: Keine Parameter

Zweck: Hiermit kann die Anzahl Fenster oder Locks auf dem VLab-Screen festgestellt werden, welche nicht von VLab selbst stammen. Erhält man einen Wert > 0, kann der VLab-Screen nicht geschlossen werden. Bei QUIT oder einem Wechsel des Screen-Typs bleibt der Screen somit geöffnet! Ein Hintergrund-Task überwacht dann den Screen und schließt ihn automatisch, wenn das letzte Visitor-Window diesen Screen verlassen hat.

RESULT: Anzahl fremder Fenster oder Locks auf dem VLab-Screen

---

Kommando: HARLEQUIN



Schablone: LACE/S, NONLACE/S, PAL/S, NTSC/S, AUTO/S,  
CLEAR/S, SCREEN/N

Zweck: Steuerung der Harlequin-Grafikkarte. Es kann gewählt werden, ob die Ausgabe im Interlace-Modus mit 15.625 kHz erfolgen soll (für 1084 Monitor oder Videogeräte) oder ohne Interlace mit 31 kHz (VGA- oder Multiscan-Monitor).

Weiterhin kann die Art der bei der Darstellung benutzten Fernsehnorm gewählt werden, bei AUTO wird abhängig vom Bild automatisch umgeschaltet. Achtung, die Karte unterstützt nicht beide Normen!

Da die Harlequin-Karte mehrere Screens unterstützt, kann die gewünschte Bildschirmseite über den Parameter SCREEN gewechselt werden. Höhere Nummern als 0 und 1 sprechen Screens auf weiteren Karten an.

Beispiele: HARLEQUIN LACE

Die Bildausgabe erfolgt in Interlace passend für z. B. einen 1084 Monitor.

#### HARLEQUIN AUTO

Bei der Ausgabe wird bei Bedarf automatisch zwischen PAL und NTSC gewechselt.

#### HARLEQUIN CLEAR 1

Nach Umschalten auf Screen 1 wird dieser gelöscht. Screen 0 bleibt unverändert.

RC: ERROR, wenn harlequin.library nicht geöffnet werden konnte  
ERROR, wenn der Screen nicht geöffnet werden konnte

---

Kommando: HELP

Schablone: COMMAND/F

Zweck: Zum angegebenen Kommando einen Hilfetext ausgeben.

RC: FATAL, wenn angegebenes Kommando nicht existiert  
ERROR, wenn keine Hilfedatei gefunden  
WARN, wenn keine Hilfe verfügbar

RESULT: Hilfetext, sofern verfügbar

---

Kommando: HUE ⊗

Schablone: LEVEL/A/N

Zweck: Mit diesem Kommando kann der Farbton bei einer NTSC-Videoquelle variiert werden. Im Pal-Modus muß HUE auf 0 stehen!  
(128 bis 255 entspricht -128 bis -1)

RC: FATAL, wenn LEVEL < -128 oder > 255

---

Kommando: HUESLIDER ⊗

Schablone: ON/S,OFF/S

Zweck: Bestimmt, ob im Einlesefenster ein Slider für den HUE-Wert vorhanden sein soll oder nicht (nur bei NTSC sinnvoll).

---

Kommando: IFR

Schablone: START/S,STOP/S,CONT=CONTINUE/S

- Zweck: Mit diesem Kommando wird IFR gestartet, unterbrochen oder nach einer Unterbrechung wieder fortgesetzt.
- 
- Kommando: IFRCONVERTMACRO ⊗
- Schablone: FILENAME
- Zweck: Bestimmen und Laden eines Makros, das nach Ende eines IFR aufgerufen werden soll.
- 
- Kommando: IFRFRAME
- Schablone: Frame/N,SET/S,CLEAR/S,ALL/S
- Zweck: Der Befehl IFRFRAME ist nur über ARexx erreichbar und ermöglicht die direkte Kontrolle über einzelne bereits eingelesene Bilder.
- Beispiele: IFRFRAME ALL SET  
Alle Bilder werden in der internen Liste als bereits fertige eingelesen markiert.
- IFRFRAME 15 CLEAR  
Bild 15 wird als noch nicht eingelesen markiert.
- Wird nach Senden dieser zwei Befehle „IFR CONTINUE“ gesendet, wird Bild 15 eingelesen.
- 
- Kommando: IFRFRAMESKIP
- Schablone: NUMBER/A/N
- Zweck: Wahl der Schrittweite, mit der IFR die Bilder einlesen soll.
- Beispiele: IFRFRAMESKIP 1  
Es wird jedes Bild eingelesen, d.h. 25 Bilder pro Sekunde (30 bei NTSC).
- IFRFRAMESKIP 5  
Es wird jedes 5. Bild eingelesen, also fünf Bilder pro Sekunde (sechs bei NTSC).
- 
- Kommando: IFRLENGTH ⊗
- Schablone: NUMBER/A/N

Zweck: Festlegung, wieviele Bilder bei der automatischen Digitalisierung einer IFR-Sequenz gespeichert werden sollen.

---

Kommando: IFRMODE ⊗

Schablone: COLOR/S,BW/S,HIRES/S,LORES/S,LACE/S,NONLACE/S

Zweck: Mit diesem Befehl wird die Einstellung des IFR-Modus vorgenommen.

Beispiele: IFRMODE COLOR NONLACE LORES

Die IFR-Bilder werden farbig digitalisiert. Außerdem wird nur ein Halbbild (NONLACE) und nur jedes zweite Pixel (LORES) gespeichert.

---

Kommando: IFRNAME ⊗

Schablone: FILENAME

Zweck: Bestimmen des Basisnamen für IFR-Sequenzen. An diesen Namen wird für jedes Bild ein Punkt gefolgt von einer vierstelligen Dezimalzahl angehängt.

---

Kommando: IFRREWINDMACRO ⊗

Schablone: FILENAME

Zweck: Bestimmen und Laden eines Makros, das nach Ende eines IFR-Durchlaufs aufgerufen werden soll.

---

Kommando: IFRSETKEYFRAME

Schablone: Keine Parameter

Zweck: IFR starten und ein neues Referenzbild bestimmen.

---

Kommando: IFRSETSCENESTART

Schablone: Keine Parameter

Zweck: Einige Zeit hinter dem Referenzbild einen Szenenstart bestimmen.

---

Kommando: IFRTOLERANCE ⊗

Schablone: Level/A/N

Zweck: Einstellen der zulässigen Toleranz bei der Referenzbild-Erkennung.

---

Kommando: IFRTRIGGERLEVEL ⊗

Schablone: Level/A/N

Zweck: Einstellen der gewünschten Triggerschwelle für die Referenzbild-Erkennung.

---

Kommando: IFRTRIGGERLINE ⊗

Schablone: Line/A/N

Zweck: Einstellen der gewünschten Triggerzeile für die Referenzbild-Erkennung.

---

Kommando: ILBMCOMPRESSION ⊗

Schablone: ON/S,OFF/S

Zweck: Durch diese Funktion wird festgelegt, ob Dateien im Amiga-IFF-ILBM-Format beim Abspeichern gepackt werden sollen oder nicht.

---

Kommando: INPUT ⊗

Schablone: NUMBER/A/N

Zweck: Dieser Befehl ermöglicht das Umschalten zwischen den im Rechnersystem vorhandenen VLab-Eingängen. Sind mehrere Geräte eingebunden, werden deren Anschlüsse an Hand der Eingangsnummer unterschieden (Belegung siehe Seite 16).

Beispiele: INPUT 1

Eingang 1 der ersten im System eingebundenen Karte wählen.

INPUT 6

Eingang 3 der zweiten im System eingebundenen Karte wählen.

RC: ERROR, wenn Eingangsnummer nicht vorhanden

FATAL, wenn Eingangsnummer  $\leq 0$  oder  $> 255$

---

Kommando: LANGUAGE ⊗

Schablone: NAME,DEFAULT/S,INTERNAL/S

**Zweck:** Auch schon unter Workbench 2.04 unterstützt *VLab* verschiedene Sprachen. Über dieses Kommando kann ein Wechsel der Sprache vorgenommen werden. Ist die gewählte Sprache nicht verfügbar, erfolgt eine Fehlermeldung.

**Beispiele:** LANGUAGE deutsch

Es sollen die deutschen *VLab*-Texte benutzt werden.

LANGUAGE DEFAULT

Die normalerweise von der Workbench benutzte Sprache soll auch von *VLab* benutzt werden

LANGUAGE INTERNAL

Es sollen keine externen Texte benutzt werden. Dies spart etwa 25 KByte Speicher ein. Die Programmoberfläche ist dann aber auf englisch, und es stehen keine Help-Texte zur Verfügung.

---

**Kommando:** LOADMACRO

**Schablone:** FILENAME

**Zweck:** Mit diesem Kommando wird der Dateiname eines zu ladenden Makros festgelegt. Aus Speicher- und Geschwindigkeitsgründen erfolgt hiermit noch kein Laden. Das Makro wird bei der ersten Benutzung automatisch geladen. Die in die Makroliste aufgenommenen Makronamen werden zusammen mit den anderen Einstellungen abgespeichert. *VLab*-Makros sind Textdateien und müssen mit einem #*VLab* beginnen. Wird bei der Ausführung des Makros kein #*VLab* gefunden, wird versucht, die Datei als ARexx-Skript, Batch-File oder ausführbares Programm zu starten.

**Beispiele:** LOADMACRO *VLab:Macros/MyMacro.vm*

Das Makro *MyMacro.vm* aus dem Verzeichnis *VLab:Macros* wird in die Makroliste aufgenommen.

LOADMACRO

Es erscheint ein File-Requester, mit dem man ein Makro aussuchen kann.

**RC:** WARN, bei Abbruch im File-Requester  
ERROR, wenn Makro nicht existiert

---

**Kommando:** LOADSETTINGS

Schablone: NAME,DEFAULT/S

Zweck: Laden der *VLab*-Einstellungen. Als da wären: CheckMarks im Menü; Festlegung, welche Fenster geöffnet sein sollen usw.

Beispiele: LOADSETTINGS DEFAULT

Es werden Standard-Einstellungen benutzt.

LOADSETTINGS FileName

Die Einstellungen aus der Datei "FileName" werden benutzt.

LOADSETTINGS

Wird kein Parameter angegeben, erscheint ein File-Requester, mit dem man die Datei mit den Einstellungen angeben kann.

RC: WARN, bei Abbruch im File-Requester

---

Kommando: LOCKGUI



Schablone: ON/S,OFF/S

Zweck: Dieses nur über ARexx verfügbare Kommando schaltet die sonstigen Eingabemöglichkeiten (Menü, Gadgets, etc.) ein oder aus.

Beispiele: LOCKGUI ON

*VLab* kann solange nicht mehr über die Programmoberfläche bedient werden, bis über ARexx das Kommando LOCKGUI OFF erfolgt. Selbst ein Beenden von *VLab* ist bei eingeschaltetem LOCKGUI nur noch über ARexx möglich. Damit man *VLab* auch bei einem abgebrochenen ARexx-Skript noch ordnungsgemäß beenden kann, besteht die Möglichkeit, die Blockierung von der Programmoberfläche aus aufzuheben. Dabei sollte man jedoch bedenken, daß eine Blockierung in der Regel einen triftigen Grund hat. Z.B. hat sich das ARexx-Skript einen Zeiger auf die Bild-Daten geholt und unterbindet mittels LOCKGUI ON eine Freigabe des zugehörigen Speichers durch *VLab*.

---

Kommando: LOCKRETINASCREENMODE



Schablone: ON/S,OFF/S

Zweck: Solange diese Option aktiviert bleibt, benutzt *VLab* nur noch die Screen-Modi der *Retina*, welche beim Einschalten der Fixierung aktuell waren. Änderungen mit *Retina-ScreenMode* werden dann erst wieder nach Aufhebung der Fixierung beachtet.

---



Kommando: LUMINANCEFILTER ⊗

Schablone: ON/S,OFF/S

Zweck: Mittels dieses Kommandos wird der Luminanzfilter auf der *VLab*-Hardware ein- oder ausgeschaltet.

Beispiele: LUMINANCEFILTER ON  
Der Filter wird eingeschaltet.

---

Kommando: MACRO

Schablone: NAME/A

Zweck: Start eines *VLab*-Makros (siehe Seite 108). Wird das zugehörige File nicht als *VLab*-Makro identifiziert, wird es (je nach Typ) als ARexx-Skript, als Batch-Datei oder als Programm ausgeführt.

RC: RC des letzten Makro-Kommandos,  
FATAL, bei vorzeitigem Abbruch,  
ERROR, wenn Makro nicht existiert.

---

Kommando: MEMORYMODE ⊗

Schablone: AUTO/S,BLOCK/S,SCATTERED/S

Zweck: Festlegung der Art der Speicherverwaltung für die Bildpuffer von *VLab*.

Beispiele: MEMORYMODE AUTO  
Es wird automatisch die günstigste Speicherverwaltung gewählt.

MEMORYMODE BLOCK

Der benötigte Speicher für die Bildpuffer muß immer am Stück verfügbar sein. Nur in diesem Modus kann über das Kommando YUV-STATUS die Adresse des Bildpuffers gelesen werden.

MEMORYMODE SCATTERED

In diesem Modus wird der Speicher für Bilddaten immer aus möglichst kleinen Stücken zusammengesetzt. Vorteil: große Stücke bleiben frei für andere Programme. Nachteil: das Anfordern und Freigeben des Speichers dauert etwas länger.

---

Kommando: MINIMIZE NOISE ⊗

Schablone: ON/S,OFF/S,LEVEL/K/N

Zweck: Hiermit wird das Verfahren zur Rauschverminderung gesteuert. Dabei kann bei Standbildern durch Mehrfachdigitalisierung statistisch verteiltes Rauschen „weggemittelt“ werden.

Die Anzahl der Mittelungen kann über den Parameter LEVEL verändert werden. Der Wert für LEVEL darf im Bereich von 2 bis 99 liegen, wobei vier bis acht Aufnahmen zu empfehlen sind.

Beispiele: MINIMIZE NOISE OFF

Die Rauschverminderung wird ausgeschaltet.

MINIMIZE NOISE ON LEVEL 6

Die Funktion wird eingeschaltet und gleichzeitig bestimmt, daß bei folgenden Digitalisiervorgängen jeweils sechs Bilder eingelesen und gemischt werden sollen.

---

Kommando: MONITOR



Schablone: ON/S,OFF/S

Zweck: Starten oder Beenden der Monitorfunktion.

Beispiele: MONITOR ON

Die Monitorfunktion wird gestartet.

---

Kommando: MONITORLOCK



Schablone: AUTO/S,ON/S,OFF/S

Zweck: Interruptsteuerung des Monitorprozesses bestimmen.

Beispiele: MONITORLOCK AUTO

Der Monitorprozeß sperrt die Interrupts bei Einlesen des Videosignals nur, wenn der VLab-Screen im Vordergrund ist.

MONITORLOCK ON

Der Monitorprozeß sperrt die Interrupts grundsätzlich beim Einlesen des Videosignals.

MONITORLOCK OFF

Die Interrupts werden auf keinen Fall gesperrt. Dies kann jedoch zum vereinzelt Durchlaufen des Bildes im Monitorfenster führen.

---

Kommando: MONITOROUTPUT ⊗

Schablone: SMALLWINDOW/S,LARGEWINDOW/S  
RETINALORES/S,RETINAHIRES/S,RETINACOLOR/S

Zweck: Dieser Befehl ermöglicht die Festlegung der Betriebsart für den Monitor. Ohne *Retina*-Grafikkarte ist lediglich die Umschaltung der Größe des Monitorfenster auf „klein“ oder „groß“ möglich. Ist der Monitor auf „groß“ geschaltet, ist die Bildwiederholrate natürlich geringer, es sind jedoch mehr Einzelheiten erkennbar.

Besitzer der o.g. Grafikkarte können die Monitorausgabe in einen *Retina*-Screen umlenken. Dazu stehen drei verschiedene Auflösungen bereit.

Beispiele: MONITOROUTPUT LARGEWINDOW

Die Monitorwiedergabe geschieht in einem großen Fenster auf dem Programm- bzw. Workbench-Screen. Dabei wird die auf diesem Screen vorhandene Farbpalette benutzt.

MONITOROUTPUT RETINAHIRES

Die Monitorfunktion verwendet eine *Retina*-Grafikkarte (falls vorhanden). Die Darstellung erfolgt in hoher Auflösung in 256 Graustufen.

---

Kommando: MONITORPRI ⊗

Schablone: HIGH/S,LOW/S

Zweck: Die Priorität des Monitorprozesses kann umgeschaltet werden.

---

Kommando: MONITORSIZE ⊗

Schablone: LARGE/S,SMALL/S

Zweck: Dieses Kommando ist nur noch aus Kompatibilitätsgründen vorhanden. Man sollte MONITOROUTPUT verwenden!

---

Kommando: MONITORSYNC ⊗

Schablone: AUTO/S,ON/S,OFF/S

Zweck: Wenn AUTO oder ON eingestellt ist, versucht *VLab* ein Flackern bei der Darstellung des Monitorbildes zu unterdrücken. Normalerweise liefert AUTO die besten Ergebnisse (siehe auch BLITMODE).

---

Kommando: MOVEWINDOW

Schablone: SCAN/S, CONVERT/S, COLOR/S, SELECTSOURCE/S,  
DEFINESOURCE/S, PREVIEW/S, SEQUENCE/S, IFR/S,  
MONITOR/S, COMMAND/S, LEFTEDGE/A/N, TOPEDGE/A/N

Zweck: Positionieren des spezifizierten *VLab*-Fensters. Es kann nur genau ein Window als Parameter angegeben werden.

Beispiele: MOVEWINDOW SCAN 200 100

Das Fenster, welches zur Steuerung des Einlesevorganges dient, wird nach Position 200,100 verschoben.

MOVEWINDOW 20 100

Läuft *VLab* auf der Workbench, kann hiermit das kleine Hauptfenster positioniert werden. In diesem Beispiel wird es nach Position 20,100 verschoben.

RC: WARN, das angegebene Fenster ist nicht geöffnet, beim nächsten Öffnen werden trotzdem die neuen Werte benutzt.

---

Kommando: NOISEFILTER



Schablone: OFF/S, LIGHT/S, MEDIUM/S, STRONG/S

Zweck: Der Rauschfilter kann auf vier verschiedene Einstellungen gebracht werden. Der Filter arbeitet nur, wenn FILTERWEIGHT nicht OFF ist.

Beispiele: NOISEFILTER LIGHT

Es findet nur eine geringe Rauschfilterung statt.

---

Kommando: OPEN

Schablone: FILENAME, FORCE/S

Zweck: Laden eines *VLab*-Bildes. Neben den normalen YUV-Bildern werden auch Bilder in dem speziellen *VLab*-Format gelesen, wie es aus Geschwindigkeitsgründen von der Sequenz-Funktion benutzt wird.

Beispiele: OPEN Bild

Die Datei "Bild" wird geladen.

OPEN

Ein File-Requester zum Laden eines YUV-Bildes wird geöffnet.

RC: WARN, wenn wegen der Sicherheitsabfrage abgebrochen wurde  
WARN, wenn der ASL-Requester abgebrochen wurde  
ERROR, wenn Speichermangel  
ERROR, wenn Load nicht funktionierte

---

Kommando: OPENWINDOW ⊗

Schablone: SCAN/S, CONVERT/S, COLOR/S, SELECTSOURCE/S,  
DEFINESOURCE/S, PREVIEW/S, SEQUENCE/S, IFR/S,  
MONITOR/S, COMMAND/S

Zweck: Öffnen von *VLab*-Fenstern

Beispiele: OPENWINDOW SCAN  
Das Fenster zum Digitalisieren und Einlesen eines Bildes wird geöffnet.  
War das Fenster schon offen, wird es nach vorne geholt.

---

Kommando: OVERSCAN ⊗

Schablone: ON/S, OFF/S

Zweck: Hiermit kann bestimmt werden, ob sich die Vorschau- und Anzeige-  
Screens in voller (Overscan-) Größe öffnen sollen (OVERSCAN ON)  
oder nur in der gleichen Größe wie der *VLab*-Screen bzw. die Work-  
bench (OVERSCAN OFF). Im zweiten Fall kann der Bildschirminhalt  
mit der Maus in alle Richtungen verschoben werden (AUTOSCROLL).

---

Kommando: PALMODE ⊗

Schablone: ON/S, OFF/S

Zweck: Wahl der Videonorm.

Beispiele: PALMODE ON  
Der PAL-Modus wird eingeschaltet.  
PALMODE OFF  
Der NTSC-Modus wird eingeschaltet.

---

Kommando: PREVIEW

Schablone: Keine Parameter

Zweck: Eine schnell berechnete Vorschau des aktuellen YUV-Bildes wird angezeigt. Die Qualität liegt deutlich unter der des allerdings langsameren "CONVERT".

---

Kommando: PREVIEWMODE ⊗

Schablone: SCALE/S,NOSCALE/S,LACE/S,NONLACE/S,COLOR/S,BW/S

Zweck: Es kann gewählt werden, auf welche Art die schnelle Vorschau eines YUV-Bildes erfolgen soll.

Beispiele: PREVIEWMODE COLOR

Die Vorschau soll in Farbe (HAM) erfolgen.

PREVIEWMODE NOSCALE BW

Die Vorschau soll in Graustufen und unskaliert erfolgen.

---

Kommando: PREVIEWOUTPUT ⊗

Schablone: WINDOW/S,BACKGROUND/S,SCREEN/S,  
HARLEQUIN/S, RETINA/S

Zweck: Durch diesen Befehl wird das Ausgabemedium für die Vorschau festgelegt: Sie kann im Fenster, als Hintergrund, auf einem eigenen Screen oder auf speziellen Grafikkarten erfolgen.

Beispiele: PREVIEWOUTPUT WINDOW

Die Vorschau erfolgt in einem zusätzlichen Fenster auf dem Programm- bzw. Workbench-Screen.

PREVIEWOUTPUT BACKGROUND

Die Vorschau soll auf dem Hintergrund des VLab-Screens ausgegeben werden (nicht auf der Workbench möglich).

PREVIEWOUTPUT SCREEN

Die Vorschau soll auf einem eigenen Screen (entweder in HAM oder in 16 Graustufen) dargestellt werden.

PREVIEWOUTPUT RETINA

Falls vorhanden, geschieht eine 24 Bit-Darstellung auf der Macro-System-Grafikkarte *Retina*.

---

Kommando: QUIT

Schablone: FORCE/S

Zweck: Beenden von *VLab*. Soll *VLab* ohne Sicherheitsabfrage beendet werden, ist der Parameter **FORCE** anzugeben.

RC: **WARN**, wenn durch eine eventuelle Sicherheitsabfrage ein Beenden von *VLab* verhindert wurde.

---

Kommando: **READ**

Schablone: **COMMAND/A**

Zweck: Dieser Befehl ermöglicht das Auslesen der Einstellungen, die bei bestimmten Kommandos vorgenommen werden können. Solche Befehle sind in dieser Liste mit dem Symbol ⊗ gekennzeichnet.

Beispiele: **READ SAVEICONS**

Liefert **ON** oder **OFF**, je nach Stellung des CheckMarks im Menü.

RC: **ERROR**, angegebenes Kommando existiert nicht oder es ist keine Rückgabe vorgesehen

RESULT: Vom angegebenen Befehl abhängige Rückgabe

---

Kommando: **REQUEST**

Schablone: **TEXT/A,TITLE/K,GADGETS/K**

Zweck: Darstellung eines Requesters auf dem *VLab*-Screen aus einem **ARexx**-Skript heraus.

Beispiele: **REQUEST "Dies ist ein Requester"**

Es erscheint ein Requester, in dem der obige Text steht.

**REQUEST "Frage?" TITLE "Test" GADGETS "ja | nein"**

Es erscheint ein Requester, in dem der Text "Frage?" steht. Der Titel lautet "Test". Außerdem gibt es noch zwei Gadgets mit den Texten "ja" und "nein".

RESULT: Nummer des Gadgets "1", "2", ..., "n-1", "0"

---

Kommando: **SAFETYLEVEL**



Schablone: **LOW/S,MEDIUM/S,HIGH/S**

Zweck: Mit diesem Kommando kann festgelegt werden, wieviele Sicherheitsabfragen erfolgen sollen.

Beispiele: SAFETYLEVEL LOW  
Bei QUIT, CLEAR usw. erfolgen keine Sicherheitsabfragen.  
SAFETYLEVEL MEDIUM  
Sicherheitsabfragen erfolgen nur an wichtigen Stellen.  
SAFETYLEVEL HIGH  
Es erfolgen grundsätzlich Sicherheitsabfragen.

---

Kommando: SAVE

Schablone: FORCE/S

Zweck: Das aktuelle YUV-Bild wird gespeichert. War schon vorher ein Bild gespeichert, erscheint vor dem Überschreiben eine Sicherheitsabfrage, es sei denn, SAFETYLEVEL LOW ist eingestellt. Diese kann mit dem Parameter FORCE unterdrückt werden.

RC: WARN, wenn kein YUV-Bild da ist  
WARN, wenn der ASL-Requester abgebrochen wurde  
ERROR, wenn SAVE nicht funktionierte

---

Kommando: SAVEAS

Schablone: NAME,FORCE/S

Zweck: Das aktuelle YUV-Bild wird gespeichert. Existiert schon eine Datei mit dem gleichen Namen, erfolgt eine Sicherheitsabfrage, falls diese nicht abgeschaltet wurden.

Beispiele: SAVEAS Bild FORCE  
Das aktuelle YUV-Bild wird ohne Sicherheitsabfrage über ein evtl. bestehendes mit gleichem Namen gespeichert.

SAVEAS  
Vor dem Speichern erscheint ein File-Requester.

RC: WARN, wenn kein YUV-Bild da ist  
WARN, wenn der ASL-Requester abgebrochen wurde  
WARN, wenn abgebrochen wurde, weil die Datei schon existiert  
ERROR, wenn SAVE nicht funktionierte

---

Kommando: SAVEICONS



Schablone: ON/S,OFF/S



Zweck: Wahl, ob zu den *VLab*-Dateien ein Piktogramm erzeugt werden soll oder nicht.

Beispiele: SAVEICONS ON  
Es sollen Icons gespeichert werden.

---

Kommando: SAVESETTINGS

Schablone: NAME

Zweck: Mit diesem Befehl werden die *VLab*-Einstellungen gespeichert.

Beispiele: SAVESETTINGS ENVARC:VLab/VLab.Prefs  
Die Einstellungen werden im Env-Archiv (ENVARC:) im Unterverzeichnis *VLab* unter dem Namen "VLab.Prefs" gesichert.

SAVESETTINGS

Es erscheint ein File-Requester, in dem man den Pfad und den Dateinamen angeben kann, unter dem die Einstellungen gespeichert werden sollen.

RC: WARN, bei Abbruch im File-Requester  
ERROR, Fehler beim Abspeichern

---

Kommando: SCAN

Schablone: Keine Parameter

Zweck: Ein YUV-Bild wird in Echtzeit digitalisiert und danach sofort in den YUV-Puffer kopiert, von wo es dann angezeigt, umgerechnet oder gespeichert werden kann.

RC: ERROR, wenn keine Hardware vorhanden  
ERROR, wenn Speichermangel vorliegt  
WARN, wenn kein Videosignal anliegt

---

Kommando: SCANMODE



Schablone: LACE/S, NONLACE/S, BW/S, COLOR/S, HIRES/S, LORES/S

Zweck: Hiermit kann gewählt werden, auf welche Art ein Bild nach dem Digitalisieren im YUV-Puffer abgelegt werden soll.

Beispiele: SCANMODE NONLACE

Es wird nur ein Halbbild in den YUV-Puffer übernommen.

SCANMODE LACE COLOR LORES

Nach der Digitalisierung wird ein farbiges LoRes-Interlace-Bild in den YUV-Puffer übernommen.

---

Kommando: SCREENDEPTH ⊗

Schablone: DEPTH/A/N

Zweck: Einstellung der Farbanzahl für den *VLab*-Screen.

Beispiele: SCREENDEPTH 3

Es wird ein Screen mit drei Planes, also acht ( $2^3$ ) Farben geöffnet.

RC: FATAL, bei DEPTH > 8.

---

Kommando: SCREENFONT ⊗

Schablone: NAME,SIZE/N,DEFAULT/S,USER/S

Zweck: Wahl eines Zeichensatzes für die Titelleisten der *VLab*-Fenster sowie für das Menü.

Beispiele: SCREENFONT Topaz 8

Es findet der Zeichensatz „Topaz 8“ Verwendung.

SCREENFONT

Ein Font-Requester erscheint, mit dem die Wahl vorgenommen werden kann.

SCREENFONT USER

Der zuletzt über den Font-Requester gewählte Zeichensatz wird benutzt.

SCREENFONT DEFAULT

Es wird der Screen-Text-Font verwendet, der über das Font-Programm der *Workbench*-Preferences eingestellt wurde.

RC: WARN bei Abbruch im Font-Requester.

---

Kommando: SCREENMODE ⊗

Schablone: WORKBENCH/S,DEFAULT/S,HIRES/S,HIRESLACE/S,  
PRODUCTIVITY/S,USER/S

Zweck: Wahl der Art des *VLab*-Screens

Beispiele: SCREENMODE WORKBENCH

*VLab* arbeitet auf dem Workbench-Screen.

SCREENMODE DEFAULT

*VLab* arbeitet auf einem eigenen Screen, der dem der Workbench entspricht.

SCREENMODE HIRESLACE

*VLab* arbeitet auf einem eigenen HiRes-Interlace-Screen.

RC: ERROR, wenn gewählter Screen-Modus nicht verfügbar

---

Kommando: SCREENSIZE



Schablone: WIDTH/N,HEIGHT/N,DEFAULT/S

Zweck: Wahl der Größe des *VLab*-Screens

Beispiele: SCREENSIZE 640 200

Der *VLab*-Screen soll eine Breite von 640 und eine Höhe von 200 Bildpunkten haben. Dies sind die minimalen Werte.

SCREENSIZE DEFAULT

*VLab* arbeitet auf einem Screen mit Standardbreite und -höhe.

RC: FATAL, wenn WIDTH < 640 oder HEIGHT < 200.

---

Kommando: SCREENTOBACK

Schablone: Keine Parameter

Zweck: *VLab*-Screen nach hinten bringen.

RC: ERROR, wenn Screen nicht offen.

---

Kommando: SCREENTOFRONT

Schablone: Keine Parameter

Zweck: *VLab*-Screen nach vorne holen.

RC: ERROR, wenn Screen nicht auf.

---

Kommando: SELECTSOURCE

Schablone: NAME

Zweck: Hiermit kann eine der zuvor definierten Quellen-Definitionen gewählt werden. Ihr Name erscheint in der Titelleiste.

Beispiele: SELECTSOURCE Kamera  
Die Quellen-Definition mit dem Namen "Kamera" wird gewählt.  
SELECTSOURCE  
Beim Aufruf ohne Parameter erscheint ein Auswahlfenster.

RC: ERROR, wenn Quelle unbekannt

---

Kommando: SEQUENCE

Schablone: START/S,STOP/S,CONT=CONTINUE/S

Zweck: Dieses Kommando steuert den Ablauf einer Sequenzaufnahme (Fortlaufendes Digitalisieren, Einlesen und Abspeichern).

Beispiele: SEQUENCE START  
Eine Sequenzaufnahme soll mit Bild Nummer 1 beginnen.  
SEQUENCE STOP  
Die laufende Sequenzaufnahme wird unterbrochen.  
SEQUENCE CONTINUE  
Eine abgebrochene Sequenzaufnahme fortsetzen oder eine neue Sequenzaufnahme beginnen, ohne den Bildzähler zurückzusetzen.

RC: ERROR, kein Videosignal oder anderer Fehler

---

Kommando: SEQUENCEDELAY



Schablone: SECS=SECONDS/N,MINS=MINUTES/K/N,TICKS/K/N

Zweck: Hiermit wird der Zeitabstand eingestellt, in dem die Bilder bei der Sequenzaufnahme hintereinanderfolgen. Die eingestellte Zeitspanne kann natürlich nur dann eingehalten werden, wenn sie nicht kürzer als die zur Speicherung benötigte Zeit ist. Wird kein Keyword angegeben, erfolgt die Zeitangabe in Sekunden. Das Keyword TICKS steht für eine Zeitangabe in fünfzigstel Sekunden. Alle Angaben werden zu einer Gesamtzeit aufaddiert. READ SEQUENCEDELAY gibt das Keyword TICKS gefolgt von der Gesamtzeit in fünfzigstel Sekunden zurück.

Beispiele: SEQUENCEDELAY 20

Bei der Sequenzaufnahme soll alle 20 Sekunden ein Bild aufgenommen werden.

SEQUENCEDELAY MINS 3 SECS 30

Es soll alle dreieinhalb Minuten ein Bild gespeichert werden.

SEQUENCEDELAY TICKS 75

Es soll alle eineinhalb Sekunden ein Bild erzeugt werden.

RC: FATAL, wenn die Gesamtzeit 9999 Sekunden überschreitet.

---

Kommando: SEQUENCELENGTH ⊗

Schablone: NUMBER/A/N

Zweck: Festlegung, wieviele Bilder bei der automatischen Digitalisierung einer Sequenz gespeichert werden sollen.

RC: FATAL, wenn  $NUMBER \leq 0$  oder  $> 9999$

---

Kommando: SEQUENCEMODE ⊗

Schablone: COLOR/S,BW/S,HIRES/S,LORES/S,LACE/S,NONLACE/S

Zweck: Mit diesem Befehl wird die Einstellung des Sequenzmodus vorgenommen. Besonders schnell arbeitet die Sequenzfunktion im Nicht-Interlace-Betrieb und einer Skalierung von 1:2, 1:3 oder 1:4.

Beispiele: SEQUENCEMODE COLOR NONLACE LORES

Die Bilder der Sequenz werden farbig digitalisiert. Außerdem wird nur ein Halbbild (NONLACE) und nur jedes zweite Pixel (LORES) gespeichert.

---

Kommando: SEQUENCENAME ⊗

Schablone: FILENAME

Zweck: Festlegung des Dateinamens für eine Sequenz. Wird kein Name angegeben, erscheint ein File-Requester.

RC: WARN, bei Abbruch im File-Requester.

---

Kommando: SEQUENCESCALE ⊗

Schablone: SCALE/A/N

Zweck: Hiermit wird ein eventueller Verkleinerungsfaktor für die Sequenzaufnahme bestimmt.

RC: FATAL, wenn SCALE = 0 oder > 4.

---

Kommando: SETPATH

Schablone: PRESETS/K,MACROS/K,YUV/K,24BIT/K,AMIGA/K,  
DELETE/K,SEQUENCE/K,IFR/K,REWIND/K,  
CONVERT/K,USER/K

Zweck: Der Pfad für einen der File-Requester wird geändert.

Beispiele: SETPATH YUV DH0:Grafik/Bilder  
Der neue Pfad für YUV-Bilder ist „DH0:Grafik/Bilder“.

---

Kommando: SETPATTERN

Schablone: PRESETS/K,MACROS/K,YUV/K,24BIT/K,AMIGA/K,  
DELETE/K,SEQUENCE/K,IFR/K,REWIND/K,  
CONVERT/K,USER/K

Zweck: Der Pattern-String für einen der File-Requester wird geändert.

Beispiele: SETPATTERN YUV ~(#?.info)  
Im ASL-Requester für YUV-Bilder werden alle Dateien, außer solchen die mit „.info“ enden, angezeigt. War vorher kein Pattern-Gadget vorhanden, wird es hiermit gleichzeitig hinzugefügt.  
SETPATTERN AMIGA ""  
Das Pattern-Gadget im ASL-Requester zum Export von Amiga-Bildern wird entfernt. Es werden danach alle vorhandenen Dateien angezeigt.

---

Kommando: SHOWTITLE



Schablone: ON/S,OFF/S

Zweck: Festlegung, ob beim Screen für die Bildanzeige und Vorschau die Titelleiste sichtbar sein soll oder nicht.

Beispiele: SHOWTITLE ON  
Titelleiste ist sichtbar

---

Kommando: SLOWSCAN ⊗

Schablone: AUTO/S,ON/S,OFF/S

Zweck: Es kann vorkommen, daß der benutzte Amiga zu langsam ist, um ein Videobild zu digitalisieren. Dieser Fall wird bei AUTO automatisch beachtet. Mit ON und OFF kann die Automatik abgeschaltet werden.

---

Kommando: SOURCENAME ⊗

Schablone: NAME/A/N

Zweck: Bezeichnung einer Videoquelle.

Beispiele: SOURCENAME Kamera  
Die Einstellungen (Filter, Eingang, Ausschnitt etc.) erhalten den Namen "Kamera".

---

Kommando: SPECIALCONVERT ⊗

Schablone: ON/S,OFF/S

Zweck: Nach SPECIALCONVERT ON können auch Bildformate berechnet werden, die auf dem Amiga nicht angezeigt werden können.

---

Kommando: TEXT

Schablone: NOLINE/S,STRING/F

Zweck: Ausgabe einer Zeichenkette im Hilfetext-Fenster.

Beispiele: TEXT Dies ist ein Test  
Der Text: "Dies ist ein Test" wird im Hilfetext-Fenster ausgegeben.  
TEXT NOLINE Hallo  
Der Text: "Hallo" wird ohne abschließenden Zeilenvorschub im Hilfetext-Fenster ausgegeben.

---

Kommando: TRANSFERRATE ⊗

Schablone: AUTO/S,NORMAL/S,HIGH/S

Zweck: Festlegung der Transferrate über den ZorroII-Bus. In Stellung „AUTO“ stellt VLab selbsttätig fest, ob eine besonders schnelle Übertragung der Daten möglich ist und schaltet (wenn möglich) darauf um. Die anderen Parameter werden nur benötigt, falls die Automatik bei einer ausgefallenen Systemkonfiguration versagen sollte.

---

Kommando: TRUECOLOR

Schablone: Keine Parameter

Zweck: Sämtliche Werte für die Farbkorrektur werden auf die neutrale Einstellung (100% bzw. 0) gebracht.

---

Kommando: UNDO

Schablone: Keine Parameter

Zweck: Das evtl. zuvor automatisch oder manuell in einen zweiten Puffer kopierte YUV-Bild wird zurückgeholt.

RC: WARN, nach Meldung, daß der Puffer leer war  
ERROR, wenn nicht genug Speicher

---

Kommando: UNZOOMWINDOW

Schablone: SCAN/S, CONVERT/S, COLOR/S, SELECTSOURCE/S,  
DEFINESOURCE/S, PREVIEW/S, SEQUENCE/S, IFR/S,  
MONITOR/S, COMMAND/S

Zweck: Die angegebenen Fenster werden auf ihre volle Größe gebracht.

RC: ERROR, wenn gewünschtes Fenster nicht geöffnet ist  
FATAL, wenn kein Fenster angegeben

---

Kommando: VERSION

Schablone: VERSION/N, REVISION/N, FULL/S

Zweck: Test der Versions- und Revisionsnummer oder Holen des Versionsstrings. Bei Angabe einer Versionsnummer wird kein Versionsstring zurückgegeben.



Beispiele: VERSION

Versionsstring holen ohne Datum

VERSION FULL

Versionsstring holen mit Datum

VERSION 1

Test, ob die Versionsnummer größer oder gleich eins ist.

VERSION 2 10

Test, ob die Versionsnummer größer oder gleich zwei ist. Ist die Versionsnummer größer als zwei, erhält man in RC den Wert 0, ist sie kleiner, den Wert 5 (Warnung). Ist sie gleich, wird getestet, ob die Revisionsnummer größer oder gleich 10 ist. Wenn nicht, erhält man ebenfalls den Wert 5 in RC.

RC: WARN, wenn Versions- bzw. Revisionsnummer zu niedrig

RESULT: Versionsstring wenn RC=OK

---

Kommando: VIDEOSTATUS

Schablone: COLOR/S,FREQUENCY/S,LACE/S

Zweck: Der Status des Videosignals kann abgefragt werden. Ist COLOR angegeben, wird geprüft, ob sich in dem gefundenen Videosignal Farbinformationen befinden (Beim Fernsehen auch bei Schwarzweiß-Filmen üblich). Ist FREQUENCY angegeben, wird bei anliegendem Videosignal dessen Vertikal-Frequenz zurückgegeben. Bei PAL sind das in der Regel 50 Hz und bei NTSC 60 Hz. VIDEOSTATUS LACE prüft, ob es sich um ein Interlace-Signal handelt.

Beispiele: VIDEOSTATUS

Prüft, ob ein Videosignal anliegt.

VIDEOSTATUS COLOR

Prüft, ob es sich um ein Farbsignal handelt.

RC: ERROR, wenn keine Hardware vorhanden

WARN, wenn kein Videosignal anliegt

RESULT: Nur wenn RC=OK; RESULT ist dann vom Parameter abhängig:  
COLOR: „0“ bei Schwarzweiß, „1“ wenn ein Farbsignal gefunden wurde.

LACE: „0“, wenn kein Interlace vorliegt, sonst „1“.

FREQUENCY: „50“ bei PAL-50 Hz, „60“ bei NTSC-60 Hz.

---

Kommando: VLABRESET

Schablone: Keine Parameter

Zweck: Die *VLab*-Hardware erhält einen Reset. Der Amiga selbst wird davon nicht berührt.

RC: WARN, wenn nach dem Reset kein stabiles Videosignal gefunden wurde.

---

Kommando: VSYNCCORRECTION



Schablone: ON/S,OFF/S

Zweck: Ist diese Option aktiviert, arbeitet der Bildfang bei schwachen Signalquellen stabiler. Nach der Umschaltung der Signalquelle kann aber ein VLABRESET notwendig werden. VSYNCCORRECTION sollte in der Regel eingeschaltet sein.

---

Kommando: VTRMODE



Schablone: ON/S,OFF/S

Zweck: Wenn direkt von einem Videorecorder digitalisiert werden soll, kann über dieses Kommando die "Time-Base-Correction" eingeschaltet werden.

Beispiele: VTRMODE ON

Die "Time-Base-Correction" wird eingeschaltet.

VTRMODE OFF

Die "Time-Base-Correction" wird ausgeschaltet.

---

Kommando: VZOOM



Schablone: LEVEL/A/N

Zweck: Beim Einlesen eines Bildes kann dieses vertikal um den Faktor 1 bis 20 vergrößert werden. Dadurch lassen sich insbesondere Zeilen aus der vertikalen Austastlücke optisch besser untersuchen.

RC: FATAL, wenn LEVEL > 20

---

Kommando: WINDOWREFRESH



Schablone: SIMPLE/S,SMART/S

Zweck: Einstellung des Refresh-Modus der *VLab*-Fenster.

Beispiele: WINDOWREFRESH SMART  
Normaler Refresh-Modus.

WINDOWREFRESH SIMPLE

Hiermit wird der speichersparende Refresh-Modus gewählt. In diesem Modus werden die *VLab*-Fenster nicht automatisch durch Umkopieren des Fensterinhaltes erneuert, sondern dadurch, daß der Inhalt neu gezeichnet wird. Dies führt natürlich zu einem langsameren Bildschirmaufbau.

---

Kommando: WINDOWTOBACK

Schablone: SCAN/S,CONVERT/S,COLOR/S,SELECTSOURCE/S,  
DEFINESOURCE/S,PREVIEW/S,SEQUENCE/S,IFR/S,  
MONITOR/S,COMMAND/S

Zweck: Ein anzugebendes Fenster nach hinten bringen.

RC: ERROR, wenn Fenster geschlossen  
FATAL, wenn kein Fenster angegeben wurde

---

Kommando: WINDOWTOFRONT

Schablone: SCAN/S,CONVERT/S,COLOR/S,SELECTSOURCE/S,  
DEFINESOURCE/S,PREVIEW/S,SEQUENCE/S,IFR/S,  
MONITOR/S,COMMAND/S

Zweck: Ein anzugebendes Fenster nach vorne holen.

RC: ERROR, wenn Fenster geschlossen  
FATAL, wenn kein Fenster angegeben wurde

---

Kommando: WORKBENCH



Schablone: OPEN/S,CLOSE/S

Zweck: Öffnen oder Schließen des Workbench-Screens.

Beispiele: WORKBENCH CLOSE

Es wird versucht, den Workbench-Screen zu schließen. Sollte dies nicht möglich sein, z.B. wenn sich Fenster auf dem Screen befinden,

die nicht zur Workbench gehören, wird ein Fehler-Code zurückgegeben.

RC: ERROR, wenn Funktion nicht ausführbar

---

Kommando: YUVSTATUS

Schablone: WIDTH/S,HEIGHT/S,HIRES/S,LACE/S,COLOR/S,  
ADDRESS/S,U/S,V/S

Zweck: Mit diesem Kommando werden Daten über das aktuelle YUV-Bild abgefragt. WIDTH und HEIGHT geben Breite bzw. Höhe des Bildes zurück. LACE und COLOR jeweils "0" oder "1". ADDRESS gibt die Adresse der Y-Daten im Speicher an, aber nur, wenn vor dem letzten Digitalisieren oder Laden MEMORYMODE BLOCK angewählt war. U oder V können in Verbindung mit ADDRESS oder WIDTH angegeben werden. Man erhält dann die entsprechenden Angaben zu den U- bzw. V-Daten.

RC: WARN, wenn kein YUV-Bild da ist  
ERROR, wenn bei ADDRESS der Speichertyp nicht stimmt

RESULT: Nur wenn RC=OK, dann ist RESULT abhängig vom Parameter:  
WIDTH: Breite des Bildes  
HEIGHT: Höhe (Zeilen) des Bildes  
HIRES: "0" wenn Lores, "1" wenn Hires  
LACE: "0" wenn Halbbild, "1" wenn Vollbild  
COLOR: "0" wenn Schwarzweiß-Bild, "1" wenn Farbbild  
ADDRESS: Adresse der Bilddaten

---

Kommando: YUVTOLORS

Schablone: Keine Parameter

Zweck: Die horizontale Auflösung des digitalisierten YUV-Bildes wird halbiert. Da nicht einfach jeder zweite Bildpunkt genommen, sondern ein Umrechnungsalgorithmus verwendet wird, erhält man trotzdem eine beachtliche Qualität.

---

Kommando: ZOOMWINDOW

Schablone: SCAN/S, CONVERT/S, COLOR/S, SELECTSOURCE/S,  
DEFINESOURCE/S, PREVIEW/S, SEQUENCE/S, IFR/S,  
MONITOR/S, COMMAND/S

Zweck: Die angegebenen Fenster werden auf Minimalgröße gebracht (Nur Titelleiste).

RC: ERROR: Fenster ist nicht geöffnet  
FATAL, wenn kein Fenster angegeben

# Kapitel 9

## Verschiedenes

### 9.1 Die ADPro-Module

Diese Module für „Art Department Professional 2.x“ der Firma ASDG ermöglichen das Zusammenspiel zwischen ADPro und VLab. Sie sollten in das Unterverzeichnis „Loader2“ kopiert werden. Zum Betrieb werden außerdem die beiden folgenden Libraries benötigt, die sich dazu im LIBS:-Verzeichnis Ihres Systems befinden müssen:

- 1.) die „vlab.library“ und
- 2.) die „adpro.library“

#### 9.1.1 Der VLAB Image-Grabber

Die Bedienungselemente des VLAB-Loaders für ADPro 2.x funktionieren prinzipiell wie die entsprechenden Kommandos der VLab-Software (siehe Abbildung 9.1).

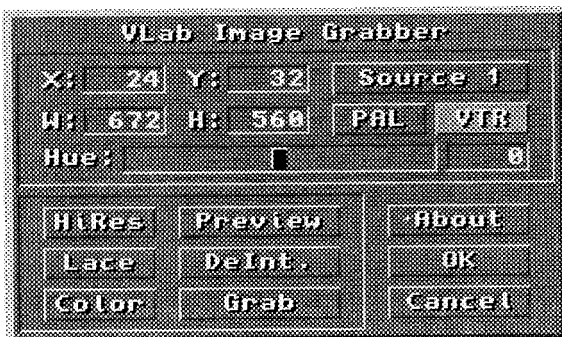


Abbildung 9.1: Direkte VLab-Digitalisierung innerhalb von ADPro

## ARexx-Interface

Template:

```
GRAB/S,  
PAL/S,NTSC/S,VTR/S,NOVTR/S,  
HIRES/S,LORES/S,LACE/S,NONLACE/S,GREY=GRAY/S,COLOR/S,  
X=LeftEdge/K/N,Y=TopEdge/K/N,W=Width/K/N,H=Height/K/N,  
Source/K/N,Hue/K/N,  
DEINTERLACE/S
```

Beispiele:

Digitalisieren eines Bildes unter Benutzung der aktuellen Einstellungen:

```
LOAD xx GRAB
```

Digitalisieren eines Bildes unter Benutzung der aktuellen Einstellungen und anschließen- dem „DeInterlace“:

```
LOAD xx GRAB DEINTERLACE
```

Verändern einiger Einstellungen:

```
LOAD xx NTSC VTR HIRES LACE COLOR Source 1 Hue 0
```

Verändern des Bildausschnittes:

```
LOAD xx X 0 topedge 0 WIDTH 720 h 480
```

Die Angabe eines beliebigen Wortes hinter „LOAD“, in den Beispielen wurde „xx“ gewählt, ist typisch für ADPro und dient als Platzhalter für den hier nicht benötigten Dateinamen.

### 9.1.2 Der YUVN Image-Loader

Das von MacroSystem für *VLab* eingeführte YUVN-Dateiformat wurde inzwischen bei Commodore als offizielle IFF-Variante registriert. Mit diesem Lade-Modul können mit *VLab* digitalisierte und abgespeicherte YUV-Bilder zur Weiterverarbeitung mit ADPro 2.x eingeladen werden. Der YUVN-Loader kann nun auch das interne *VLab*-Format einlesen (funktioniert jedoch nicht über den UNIVERSAL-Loader).

Sofort nach dem „Grabben“ bzw. Einladen werden die Bilddaten automatisch vom YUV-Format in die ADPro-interne 24 Bit-RGB-Darstellung umgerechnet.



Abbildung 9.2: Einladen von Dateien im VLab-YUVN-Format



Abbildung 9.3: Umrechnung von YUV-Bildern in das ADPro-RGB-Format



## 9.2 Tips und Fehlerhilfen

Dieser Abschnitt soll bei auftretenden Problemen Hilfestellung bieten. Leider werden Mißverständnisse oder Bedienungsfehler oft für Störungen und Fehlfunktionen gehalten. Wenn Sie den Eindruck haben, daß Ihr Exemplar von *VLab* nicht in Ordnung ist, überprüfen Sie die unten aufgeführten Punkte! Untersuchen Sie aber auch Ihre übrigen Systemkomponenten! Sollten Sie die Schwierigkeiten nicht selbst beheben können, setzen Sie sich bitte mit Ihrem Fachhändler oder mit MacroSystem direkt in Verbindung.

### **Das Programm kann die VLab-Hardware nicht erkennen.**

Eventuell liegt ein Hardware-Defekt vor. Es kann sich jedoch auch um einen Adreßkonflikt im AutoConfig-Bereich handeln. Entfernen Sie andere Platinen oder tauschen Sie die Reihenfolge der Erweiterungskarten aus. Drücken Sie alle Platinen richtig in die Slots und reinigen Sie die Goldkontakte der *VLab*-Karte mit einem Radiergummi.

### **Die VLab-Software findet das Betriebssystem nicht.**

*VLab* kann nicht mit Vorversionen des OS2.X zusammenarbeiten. Fragen Sie Ihren Commodore-Fachhändler nach einem Update! Sie können auch Teile des Betriebssystems von Ihrer *VLab*-Diskette kopieren. Insbesondere die „asl.library“ im LIBS:-Verzeichnis sollte auf neuestem Stand sein.

### **VLab empfängt kein Videosignal.**

Wahrscheinlich ist nur der falsche Videoeingang aktiviert oder der Cinch-Stecker hat keinen richtigen Kontakt. Achten Sie auch darauf, daß kein Antennenkabel direkt an *VLab* angeschlossen werden darf. Vielleicht liegt auch ein Fehler in Ihrem Verbindungskabel vor. Außerdem liefert nicht jeder Videostecker an einem Fernsehgerät das benötigte FBAS-Signal (Video out). Lesen Sie noch einmal Abschnitt 3.2. Unter Umständen müssen Sie Ihren Rechner aus- und wieder einschalten.

### **Es ist kein Betrieb von VLab/par möglich.**

Sie haben wohl vergessen, die externe Hardware vor dem Programmstart einzuschalten. Nun müssen Sie der Software das Vorhandensein von *VLab/par* nachträglich mitteilen, indem Sie die Funktion *VLab/par* einbinden aufrufen.

### **Der Monitorbetrieb funktioniert nicht richtig.**

Ist das Bild dunkel, wird kein Videosignal empfangen (siehe oben). Ist bei einem Amiga 3000/4000 nur ein wirres Geflimmer erkennbar, sollten Sie den Zustand der *VLab*-Variable „TRANSFERRATE“ überprüfen und auf „AUTO“ oder sogar „NORMAL“ setzen. Tritt starkes, großflächiges Flackern auf, ist das Monitorfenster wahrscheinlich teilweise von anderen Fenstern überdeckt. Eventuell ist auch die Einstellung von „BLIT-MODE“ ungünstig gewählt.

### **Im Bild treten Fehler auf.**

In der benutzten Quellendefinition sollte in der Regel „VSC“ und im Videorecorderbetrieb zusätzlich „VTR“ eingeschaltet sein. Treten trotzdem Versetzungen auf, sollte man die Einstellung der Fernsehnorm (PAL/NTSC) überprüfen.

Bei Farbfehlern können Sie zunächst das empfangene Signal mit einem Fernsehgerät überprüfen. Ist es in Ordnung, sollten Sie die Einstellung der Farbtreue (COLORFIDELITY, normal 254) testen. „HUE“ wird nur im NTSC-Betrieb verwendet und muß bei PAL auf null gesetzt sein. Denken Sie daran, daß ein Computermonitor eine etwas andere Farbdarstellung als ein Fernsehapparat haben kann (siehe auch Abschnitt 6.8.3).

### **Ein frühes VLab/par-Modell liefert häufig starke Farbfehler.**

Es könnte sich hier ein Serienfehler zeigen. Nehmen Sie bitte Kontakt mit dem MacroSystem-Service auf!

### **Ein Amiga mit 32 Bit-CPU kann keine Bilder digitalisieren.**

Benutzen Sie das Fenster Befehl ausführen oder das „VShell“-Makro, um das ARexx-Kommando TRANSFERRATE NORMAL einzugeben. Speichern Sie anschließend die *VLab*-Einstellungen.

### **Das Bild flackert oder hat kammförmige Verzerrungen.**

Diese Effekte sind die Folge von Bewegungen zwischen zwei Halbbildern. Wenden Sie einfach die Deinterlace-Funktion an (siehe auch Abschnitt 6.2).

### **Bei der Standbildaufnahme treten Probleme auf.**

Hakt der Mauszeiger, ist das Bild um die Hälfte versetzt oder oben gestört, sollten Sie am Videorecorder solange das Einzelbild weiterschalten, bis die Störung behoben ist.

### **Das Bild ist verrauscht oder unscharf.**

Häufig weist schon das empfangene Videosignal diese Mängel auf. Man sollte bedenken, daß das menschliche Auge bei bewegten Bildern eine „Mittelung“ vornimmt und so Bildfehler wegrechnet. Durch die HAM-Umrechnung hingegen werden diese sogar verstärkt (auf einer 24 Bit-Grafikkarte sieht das Ergebnis dann besser aus).

Sie können die Einstellmöglichkeiten bei der Quellendefinition (Gewichtung, Filter, Frequenz) nutzen, um die Digitalisierung zu optimieren.

### **Das Programm erhält nicht genügend Speicher.**

Leeren Sie den Inhalt der RAM-Disk — dort befinden sich häufig noch digitalisierte Bilder oder Sequenzen. Denken Sie daran, daß *VLab* eventuell in „RAM:ENV“ seine Einstellungen speichern will. Vielleicht ist auch die vertikale Vergrößerung „VZOOM“ ungleich eins. Ansonsten sollten Sie die Hinweise im Abschnitt 4.5 beachten.

### **Der RS 232-Betrieb arbeitet nicht mehr.**

Falls Probleme beim RS 232-Betrieb auftreten, sollten Sie „MONITORLOCK“ auf „OFF“ umstellen und ein anderes Handshaking-Protokoll benutzen. Können Sie auf den Monitorbetrieb verzichten, schalten Sie die Monitorfunktion während der RS 232-Benutzung einfach aus.

### **Das Bild flackert beim Einlesen und wenn Monitor aktiv ist.**

Sie haben einen Amiga mit 68000 oder 68010 CPU und kein oder kein freies FAST-Mem. Zusätzlich arbeitet *VLab* auf einem Screen mit acht oder 16 Graustufen. Der Amiga wird in diesem Zustand so langsam, daß zur sauberen Synchronisation des einzulesenden Videobildes die Bildausgabe des Amiga kurzzeitig abgeschaltet werden muß. Dadurch entstehen die Bildstörungen. Das Phänomen kann durch das *VLab*-Kommando „SLOWSCAN“ beeinflusst werden.

### **Die Software sucht nach „ENV:“ oder „ENVARC:“.**

Wahrscheinlich werden in Ihrer startup-sequence die „assigns“ von „ENV:“ und „ENVARC:“ nicht (korrekt) eingerichtet. Ziehen Sie Ihr Handbuch zum Betriebssystem zu Rate!

### **ARexx funktioniert nicht richtig.**

Überprüfen Sie die Installation von ARexx auf Ihrem Betriebssystem (siehe auch Seite 108). Ziehen Sie dazu Ihre OS 2.X-Anleitung zu Rate!

### **Bei Monitorbetrieb ruckelt der Mauszeiger.**

Zur exakten Synchronisation mit dem einzulesenden Bild müssen vor Einlesen eines Bildes kurzzeitig alle Interrupts gesperrt werden. Während dieser Zeit (ca. 1/50 sec) kann der Mauszeiger nicht bewegt werden, daher das Ruckeln. Der ARexx-Befehl „MONITORSYNC“ steuert das Sperren der Interrupts.

### **Der verwendete Bildausschnitt stimmt nicht.**

Weil *VLab* mehr Bildpunkte einlesen kann als die Videoquelle an Bildmaterial enthält, muß der gewünschte Bildausschnitt vom Anwender richtig eingestellt werden. Lesen Sie dazu die Abschnitte 5.2.7 und 6.4!

### **Ein PublicWindow hat falsche Farben.**

*VLab* ändert die Farbpalette für seinen Screen. Fremde Programme, die ihr Fenster auf dem *VLab*-Screen öffnen benutzen oft die falschen Farben. Die richtigen Farben werden unter OS-Release-2 in der DrawInfo-Struktur angegeben. Wer diese Angaben nicht beachtet, programmiert nicht korrekt und muß mit Fehlern wie diesem rechnen. Von alten 1.2/1.3-Programmen, die sich durch Patches auf den *VLab*-Screen bringen, soll hier gar nicht erst gesprochen werden.

### **Andere Programme können keine VLab-Dateien lesen.**

Denken Sie daran, daß das von *VLab* eingeführte YUVN-Format noch nicht von allen Programmen verstanden wird. Schwierigkeiten können auch bei den Bildbreiten entstehen (siehe Seite 99).

### **Die Menüs von VLab-Screens sind unleserlich.**

Bei der Anzeige von Bildern auf einem Amiga-Screen sind manchmal Teile des Menütextes kaum lesbar. Dieser Effekt ist bedingt durch die optimale Farbwahl bei den Bildern, die leider teilweise für die Menüdarstellung recht ungünstig ist.

## 9.3 Über Public Screens

Die *VLab*-Software öffnet einen „Public Screen“, also einen Bildschirm, den andere Anwendungen gemeinsam mit *VLab* benutzen können. Neben den eigenen Fenstern können sich also auch die Windows weiterer Programme gleichzeitig auf dem Screen von *VLab* befinden.

Sollte der *VLab*-Bildschirm nicht zum Standard-Public-Screen erklärt worden sein, müssen Applikationen, die Fenster auf *VLab* öffnen wollen, eine Zeichenkette als Kennung für den Public Screen angeben. Dieser Name ist „VLAB.n“, wobei n eine Zahl zwischen 1 und 99 ist, um bis zu 99 Screens unterscheiden zu können. Den tatsächlichen Namen erfährt man im Menüpunkt „Über VLab...“ oder mit dem AREXX-Befehl „GETPUBSCREENNAME“.

Als Beispiel sollten Sie einmal versuchen, mit dem folgenden CLI-Kommando eine gewöhnliche Shell auf dem *VLab*-Screen zu öffnen:

```
newshell CON:20/80/600/100/Title/CLOSE/SCREENVLAB.1
```

## 9.4 Internes für Programmierer

Folgende Strukturen werden von *VLab* im System angemeldet:

```
Prozesse: „VLab“  
          „VLab.n PubScreen Manager“  
          „VLab Monitor“
```

Das Hauptprogramm *VLab* erhält den Namen „VLab“. Wird *VLab* mehrfach gestartet, haben auch alle anderen Prozesse denselben Namen. Jedes gestartete *VLab*-Programm öffnet einen PublicScreen, der von einem Prozeß mit Namen „VLab.n Pub Screen Manager“ verwaltet wird. „n“ ist dabei die Nummer des entsprechenden Screens. Die Monitor-Funktion startet einen Prozeß mit Namen „VLab Monitor“. Dieser Prozeß ist nur vorhanden, solange die Monitor-Funktion in Betrieb ist.

Bei dem Port mit Namen „VLAB“ handelt es sich um den AREXX-Port von *VLab*.

Zur Verwaltung der *VLab*-Hardware wird pro benutzte Hardware eine Signal-Semaphore „VLab Hardware 00Ex0000“ (x=[9..F]) eingerichtet, über die eine quasi gleichzeitige Nutzung einer Hardware von mehreren Prozessen möglich wird.

## 9.5 Die Tastatur-Kurzbehele (Short Cuts)

Es werden alle Menüpunkte aufgeführt, die durch eine Tastenkombination direkt ansprechbar sind. Die angegebenen Tasten sind immer in Verbindung mit der rechten Amiga-Taste zu drücken.

VLab 4.0	Menüpunkt	Taste	
<b>Projekt:</b>	Laden...	O	
	Speichern	S	
	Speichern als...	A	
	Export	» Amiga	X
		24 Bit ILBM	B
		24 Bit DEEP	V
	VLab beenden	Q	
<b>Fenster:</b>	Einlesen...	1	
	Umrechnen...	2	
	Sequenz...	3	
	Film (IFR)...	4	
	Farbkorrektur...	5	
	Quelle wählen...	6	
	Quelle definieren...	7	
	Befehl ausführen...	9	
	Fenster schließen	0	
<b>Steuern:</b>	Einlesen	» Start	G
	Umrechnen	» Start	T
		Bild anzeigen	D
	Sequenz	» Start	F
	Bearbeiten	» DeInterlace	I
		YUV-Bild sichern	C
		Bild zurückholen	U
	Monitor	» Betrieb	M
		Großes Fenster	L
	Vorschau	» Vorschau anzeigen	P
<b>Einstellungen:</b>	Screen typ	» HiRes	H
	Einstellungen laden	» Temporäre	E



Scanned  
by  
10<sup>st</sup>

Taken from Amiga-Manuals-Website