

# 3D/Sequence Manual



## picCOLOR Industrial and Scientific Image Processing and Analysis

F.I.B.U.S. Dr. R. H. G. Müller Beim Elbkuurhaus 12 D-22587 Hamburg  
Tel.: 49-40-24421280 FAX: 49-40-24421282  
email: [info@fibus.org](mailto:info@fibus.org) WWW: <http://www.fibus.org>

# Inhalt

1. Einleitung zur 3D/Sequenz-Bildverarbeitung .....	3
2. Funktionsübersicht: 3D/Sequenz-Modul.....	5
3. Definition eines 3D-Bildes / einer Sequenz .....	7
4. Darstellung von 3D-Bildern/Sequenzen.....	9
4.1. Einzelbilddarstellung .....	9
4.2. Auswahl aus verkleinerter Darstellung aller Bilder .....	9
4.3. Videoplayer für 3D-Bild und Sequenz.....	9
4.4. Graphische Overlay-Funktionen für die Videodarstellung .....	10
4.5. Ausgabe von Sequenzen auf einem Videorekorder.....	10
4.6. Ausgabe als AVI-Video-File (siehe auch Kap. 6, Laden und Speichern) .....	10
5. Sequenz-Bildübernahme .....	11
5.1. Parameter für die Sequenzaufnahme.....	11
5.2. Beispiel einer Sequenzaufnahme .....	15
5.3. Sequenzaufnahme als Video-Stream direkt auf Disk.....	16
5.4. Hochgeschwindigkeitskameras .....	17
5.5. Farbbildsequenzen und Stereo-Sequenzen.....	18
5.6. Einlesen von 3D-Bildern.....	18
5.7. Einlesen mit arithmetischer/logischer Funktion.....	18
6. Ein- und Ausgabe auf Festplatte .....	19
6.1. Lade 3D/Sequenz-Bild von Festplatte.....	19
6.2. Speichere Sequenz/3D-Bild .....	20
6.3. Laden und Abspeichern von Stereo-Bildsequenzen.....	20
6.4. Abspeichern von Sequenzen im AVI-Format .....	20
7. Verarbeitung von 3D -Bildern .....	22
7.1. Kopiere 3D-Schnittbild -> Bildspeicher .....	23
7.2. 3D-Geometry .....	23
7.3. 3D-Arithmetik.....	23
7.4. 3D-Filter.....	23
7.5. 3D-Morphologie.....	23
7.6. 3D-Frequenzraumverarbeitung .....	23
7.7. 3D-Binärisierung.....	25
7.8. 3D-Vermessungsfunktionen.....	25
7.9. 3D-MACRO-Programme .....	25
8. Verarbeitung von Sequenz-Bildern.....	27
8.1. Kopiere Sequenz-Schnittbild auf den 2D-Bildspeicher .....	27
8.2. Sequence Processing Functions .....	27
8.3. Sequenzverarbeitung mit MACRO-Programm .....	28
8.4. Benutzung der 3D-Funktionen für Sequenzverarbeitung.....	31
8.5. Sequenzvermessungsfunktionen .....	31
9. Fehlermeldungen 3D/Sequenz .....	32
9.1. Fehler bei Definition der 3D/Sequenz-Bilder .....	32
9.2. Fehler bei Verarbeitung der 3D/Sequenzbilder.....	32

# 1. Einleitung zur 3D/Sequenz-Bildverarbeitung

Die Auswertungen von Bewegungssequenzen und von 3D-Bildern sind ganz besonders wichtige und komplexe Aufgaben der digitalen Bildverarbeitung. Wie ist es jedoch zu erklären, daß 3D-Modul und Sequenz-Modul zu einem einzigen picCOLOR-Zusatzmodul zusammengesetzt wurden? Was hat schließlich ein 3-dimensionales Magnetresonanz-Tomographie-Schnittbild (MR) oder Computer-Tomographie-Schnittbild (CT) eines menschlichen Kopfes mit einem schnellen Bewegungsablauf, z. B. im Sport gemeinsam? Nun - die Erklärung ist sehr einfach: ein 3D-Bild kann man sich als Serie von 2-dimensionalen Schnittbildern vorstellen. Genau dies passiert z. B. auch beim Durchfokussieren eines durchsichtigen Objektes unter einem Mikroskop: man sieht immer einen einzigen Schnitt scharf - alle anderen Schnitte verschwinden (mehr oder weniger), da sie außerhalb der Fokusebene zu liegen kommen. Man kann sich hierbei die einzelnen Schnitte des 3-dimensionalen Gegenstandes nur nacheinander anschauen - man betrachtet also ein 3D-Bild in einer zeitlichen Sequenz. Später kann man dann alle Schnitte als Bildblock hintereinander anordnen und als tatsächliches 3D-Bild betrachten - zumindest theoretisch oder unter bestimmten Randbedingungen, denn praktisch ist dies mit einem normalen Computer-Monitor wohl schwerlich machbar... Genauso gut kann man nun einen Sportbewegungsablauf, der ja normalerweise als zeitliche Sequenz abläuft, als 3D-Schnittbild betrachten - es wird ganz einfach die zeitliche Dimension in eine räumliche umgewandelt. Hier ist es dann z. B. durchaus sinnvoll, eine der beiden Raumdimensionen über der Zeit aufzutragen, so daß die Bewegung eines Objektes als Raumkurve sichtbar wird. Diese Erkenntnis, daß ein 3D-Bild und eine Sequenz aus 2D-Bildern auch in der jeweils anderen Anordnung durchaus sinnvoll angesehen werden können, kann bei der Verarbeitung der Bilddaten sehr hilfreich sein und bei der Auswertung eine große Rolle spielen.

Man sieht also, daß es gar keinen so großen Unterschied zwischen einer Sequenz, d. h. einem Zeitreihenbild und einem 3D-Bild gibt. Auch bei der Behandlung von Sequenzen und 3D-Bildern im Bildverarbeitungsrechner gibt es viele Parallelen: die Speicherung der Bilder wird immer als Serie 2-dimensionaler Schnitte vorgenommen und die Ansicht geschieht üblicherweise ebenfalls in 2-dimensionalen Schnitten. Da die 3D- und Sequenz-Funktionen damit softwaremäßig auf den gleichen Grundfunktionen aufbauen, ist es sicherlich vernünftig, die beiden Zusatzmodule 3D-Imaging und Sequenzbearbeitung zusammenzulegen. Die Verarbeitung der Bilder ist jedoch in der Regel unterschiedlich: sicherlich ist es in den wenigsten Fällen sinnvoll, bei einer Bewegungssequenz eine Kantenverstärkung in zeitlicher Richtung vorzunehmen - für ein 3D-Bild ist es jedoch völlig normal, solch einen Filter in z-Richtung anzuwenden. Wie oben schon angedeutet, kann es jedoch auch neuartige Möglichkeiten der Auswertung ergeben, wenn man die Auffassung von 3D-Bild und Sequenz ein wenig miteinander vermischt. Hier sei die Phantasie des Anwenders angeregt...

Bei den üblichen Bildsequenzen sind die Einzelbilder immer nur 2-dimensionale Bilder, da die Kamera, die bei der Aufnahme benutzt wird, üblicherweise ein 2-dimensionales Abbildungsgerät ist. Schwieriger wird es, wenn man den zeitlichen Ablauf eines 3-dimensionalen Bewegungsablaufes erfassen möchte: man erhielte ein 4-dimensionales Raum-Zeit-Bild. Die Simulation eines solchen Bildes im Rechner ist sicherlich kein Problem und könnte auf speziellen Wunsch programmiert werden. Die bildliche Darstellung und noch viel mehr die Bildaufnahme wird jedoch mit größeren Problemen verbunden sein - mit einer einfachen Kamera wird man da nicht mehr auskommen. Bei kleinen durchsichtigen Objekten unter dem Mikroskop könnte man dazu etwa Fokussierreihen zu verschiedenen Zeiten anfertigen. Mit den neu auf den Markt gekommenen High-Speed-Kameras wird es bis zu einem bestimmten Grad auch möglich sein, komplette 3-D-Bilder von Strömungsvorgängen aufzunehmen: bei einer Aufnahmegeschwindigkeit von z. B. 5000 Bildern/Sekunde mit 512\*512 Pixeln könnte man hierbei mit einer bewegten Lichtschnittbeleuchtung (z. B. mit Drehspiegel) 3-D-Bilder der Auflösung 512\*512\*512 Voxel (von: Volume Pixel Element) mit 10 Bildern/Sekunde aufnehmen. Das ist doch schon etwas! Der hierzu notwendige Speicherbedarf von 1280 MByte/s ist allerdings auch nicht zu verachten!

Zur 3-dimensionalen Untersuchung könnte man übrigens auch mit zwei stereometrisch angeordneten Kameras Stereo-Sequenzen aufnehmen und auswerten. Die Aufnahme solcher Stereo-Sequenzen ist mit dem 3D/Sequenz-Zusatzmodul prinzipiell möglich, wenn sich Kameras und Bildeinlesekarten dafür eignen. Für die Auswertung solcher Stereo-Sequenzen steht z. B. das 3D-Position-Tracking Zusatzmodul zur Verfügung. Zur Ermittlung von 3D-Formen undurchsichtiger Objekte und deren zeitlicher Änderung können z. B. Meßmethoden mit Kamera und Lichtstreifenprojektoren dienen, die die gekrümmte Objektfläche über die verzerrte Projektion der Lichtstreifen auswerten. Dies kann je nach Anwendung mit dem FCM-Zusatzmodul oder mit dem Interferometrie-Zusatzmodul durchgeführt werden. Hiermit erhält man jedoch lediglich die Oberflächenstruktur. In der Magnet-Resonanz-Tomographie kommen immer schnellere Systeme auf den Markt, so daß es wohl bald möglich sein wird, hiermit 3-dimensionale Zeitreihenbilder zu erzeugen.

Doch nun zurück zum einfachen 3D/Sequenz-Bild: Die Speicherung von 3D-Bildern oder Sequenzen benötigt in der Regel einen größeren Speicherausbau und die Auswertung derselben eine entsprechende Rechenzeit. Hierdurch kann es notwendig werden, spezielle Programmoptimierungen durchzuführen oder auf schnellere Prozessoren auszuweichen. Als Beispiel sei hier der oben schon angedeutete Speicherbedarf eines 512\*512\*512-Bildes mit 128 MByte und der eines 10 Sekunden langen Bewegungsablaufes in einer einfachen Bildauflösung von 1280\*1024 Bildpunkten bei 20 Hz Bildfrequenz mit 256 MByte angegeben. Und bei Hochgeschwindigkeits-

systemen kann es auch noch deutlich mehr werden: z. B. bei einer Bildauflösung von 1280\*1024 Pixeln und einer Aufnahme­frequenz von 500 Bildern/Sekunde wären dies 660 Mbyte in nur einer Sekunde! Je nach Funktionsaufwand lassen sich solche Datenmengen auch mit sehr schnellen Prozessoren kaum noch in ein paar Sekunden bearbeiten. Man wird sich also auf längere Auswertezeiten einstellen müssen. Jede Optimierung der Funktionen ist hier natürlich sehr hilfreich...

Die Funktionen zur 3D-Bild/Sequenz-Definition, zum Bildeinlesen oder -abspeichern, zur Bildaufnahme und zum Bildbetrachten sind in der Software fest installiert. Ebenso eine Reihe von 3D-Funktionen bis hin zur 3D-Fouriertransformation sowie Filterung und Arithmetik im Frequenzbereich. Andere Funktionen – speziell komplexe Funktionsabläufe – werden jeweils als MACRO-Programme implementiert. Dies sind Bildverarbeitungsmethoden, die über den eingebauten MACRO-Interpreter automatisch oder halbautomatisch abgearbeitet werden. Beispiele sind als Quellcodes beige­fügt und lassen sich damit leicht vom Anwender ändern und für spezielle Zwecke anpassen. Sequenzbearbeitungsfunktionen wurden nur sehr wenige fest implementiert, da diese zum größten Teil auf den normalen 2-dimensionalen Funktionen basieren und damit über einfache MACRO-Funktionen programmiert werden können. Einige beispielhafte MACRO-Funktionen zur Sequenz-Bearbeitung werden hier im Handbuch beschrieben.

Die Reihe der verfügbaren Funktionen wird ständig erweitert. Im folgenden Kapitel ist eine Liste der z. Zt. verfügbaren Funktionen angeführt.

## 2. Funktionsübersicht: 3D/Sequenz-Modul

Da 3D- und Sequenz-Bilder oft riesige Datenmengen bedeuten, wird auch die Verarbeitung recht aufwendig sein, zumindest zeitaufwendig. Daher sollten diese Funktionen gut optimiert werden. Dies bedeutet jedoch auch, daß diese Funktionen in der Regel nicht völlig allgemeingültig, d. h. mit vielen variablen Parametern programmiert werden können, sondern genau an die Erfordernisse angepaßt werden müssen. Jede zusätzliche Rechenoperation, die dann mit jedem Voxel durchgeführt werden muß, kann die Ausführungszeit erheblich verlängern. Daher sind nur recht wenige allgemeine Funktionen implementiert sowie einige spezielle Funktionen, die für bestimmte Projekte benötigt wurden. Wenn Sie also spezielle Funktionen wünschen, erkundigen Sie sich beim Entwicklungsteam, ob es diese Funktionen eventuell bereits gibt oder in welchem Zeitraum sie implementiert werden können.

Definition eines 3D-Bildes oder einer Sequenz

- Set x-size, y-size, and z-size (oder Anzahl Bilder/Sequenz)
- Allocate 3D/Sequence-Buffer
- Deallocate/Close 3D/Sequence-Buffer

Darstellung von 3D-Bildern oder Sequenzen

- Videoplayer für 3D-Bildschnitte oder Sequenz
- Videoplayer für speziellen Real-Time-Video-Stream
- Pixelinspektionsfunktion für xyz-Koordinaten und Grauwert.
- Auswahl aus einer Serie verkleinert dargestellter Schnittbilder
- Anzeige von xy-, yz- oder zx-Schnitt eines 3D-Bildes
- Anzeige von xy-, xt-, yt-Schnitt einer Sequenz (t=time)

Bildübernahme von 3D-Bildern und Sequenzen

- Einzelbildaufnahme (Single Shot) mit Mausklick
- Kontinuierliche Sequenzaufnahme
- Benutzung des Sequenzspeichers als unendlichen Ring-Speicher
- Aufnahme im Interlaced- oder im Noninterlaced-Modus (für schnelle Bewegungen bei Interlaced-Kameras)
- Aufnahme von Bildsequenzen als Real-Time-Video-Stream oder in Einzelbildern direkt auf die Festplatte

Ein- und Ausgabe auf Festplatte

Als Speicherformat stehen alle im Basisprogramm implementierten Bildformate zur Verfügung. In der Regel wird man das bekannte TIF-Format oder das bei den großen Datenmengen sehr sinnvolle, da stark komprimierende JPEG-Format benutzen. Eine Kompression auf 10 bis 20% der ursprünglichen Datenmenge ist bei immerhin kaum sichtbarer Bilddegradation möglich.

- Lade Sequenz oder 3D-Bild von Festplatte
- Speichere Sequenz oder 3D-Bild auf Festplatte
- Speichern einer Sequenz im (z. Zt. noch unkomprimierten) AVI-File-Format
- Laden von Real-Time-Video-Streams
- Konvertieren eines Real-Time-Video-Streams in das AVI-Format

3D-Funktionen

- Kopiere 3D-Schnittbild in den normalen 2D-Bildspeicher und zurück in den 3D-Speicher
- Kopiere xy, xz, zx-Schnitt des 3D-Bildes in den Arbeitsspeicher
- Arithmetische Funktionen für ein 3D-Bild oder zwei 3D-Bilder:
  - Löschen eines 3D-Bildes
  - Invertieren
  - Addieren oder Subtrahieren einer Konstanten
  - Multiplizieren einer Konstanten
  - Scaling, Quantify, Power
  - Logische Funktionen mit Konstanten (AND, OR, XOR, NAND, Shift, usw.)
  - Kopieren eines 3D-Bildes auf ein anderes
  - Logische Funktionen zwischen zwei 3D-Bildern (AND, OR, XOR usw.)
  - Arithmetische Funktionen mit zwei 3D-Bildern (MAX, MIN, SUB, ADD, MUL, DIV, usw.)
  - Energieberechnung von zwei Bildern, zyklisches Subtrahieren von zwei Bildern
- 3D-Filter:
  - Laplace
  - Sobel (Sobel-Hx, Sobel-Hy, Sobel-Hz und auf Wunsch Sobel-Betrag)
  - Average

- Gauss
- Sharpen, Sharpen more (= Laplace + Originalbild)
- Rank-Filter (3\*3\*3 oder 5\*5\*5)
- Median-Filter (3\*3\*3 oder 5\*5\*5)
- 3D-Morphologie:
  - Erosion (3\*3\*3 oder 5\*5\*5)
  - Dilataion (3\*3\*3 oder 5\*5\*5)
- 3D-Fouriertransformation:
  - 3D-FFT-Transformation
  - 3D-IFFT (inverse Transformation)
  - Shift DC: Schiebe DC-Anteil bei der FFT-Transformation in die Mitte
  - Transformation mit oder ohne Hamming Window
  - Berechnung von Amplitudenspektrum, Phasenspektrum, Realteil oder Imaginärteil
  - Umrechnen von Real/Imaginär in Phasen/Amplituden-Spektrum
  - Tief-, Hoch- und Bandpassfilter
  - Gaussian Bandpass, Gaussian Smooth
  - Gabor Filter
  - Difference of Gaussian
  - Inverse oder Wiener Filter
  - 3D-FFT-Buffer-Arithmetik:
    - Clear active 3D-FFT-Buffer
    - Add(/Sub) Const.
    - Mul(/Div) Const.
    - Konjugiert Komplex
    - Inverse Complex
    - Normalize Complex
    - Active = Active + Second (auch getrennt Real/Imaginärteil)
    - Active = Active - Second
    - Active = Active \* Second (auch getrennt Real/Imaginärteil)
    - Active = Active / Second
    - Auto correlation with active buffer
    - Cross Correlation of active and second buffer
  - Erzeugung einer theoretischen 3D-Polynom-OTF (Optische Transferfunktion)
  - Shift-DC (DC-Anteil in der Bildmitte oder in der vorderen linken oberen Ecke)
- 3D-Binärisierung:
  - Threshold
  - Band (lower and upper threshold)
- 3D-Vermessungsfunktionen:
  - 3D-Volumenbestimmung binärisierter Bilder (über MACRO-Programm)
  - 3D-Punktvermessung
  - 3D-Histogram in z-Richtung

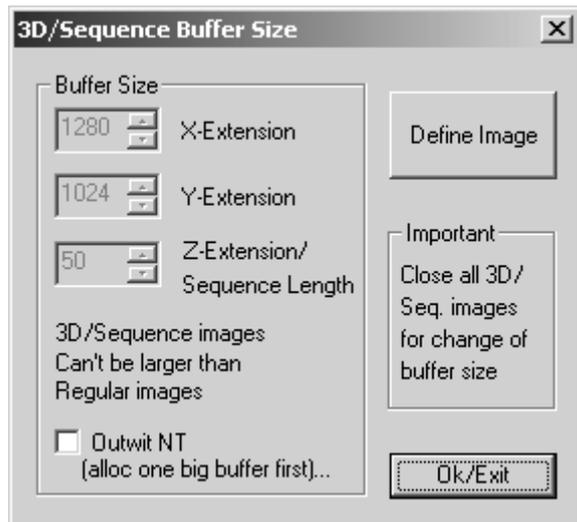
#### Sequenzverarbeitungsfunktionen:

- Kopiere xy, xt, yt-Schnittbild vom normalem 2D-Bildspeicher in den Sequenzspeicher und zurück
- Arithmetische Funktionen mit einer Sequenz oder mit zwei Sequenzen:
  - Invertieren
  - Addieren oder Subtrahieren einer Konstanten
  - Multiplizieren einer Konstanten
  - Kopieren einer Sequenz in einen anderen Sequenzspeicher
  - Logische Funktionen zwischen zwei Sequenzen
  - Arithmetische Funktionen mit zwei Sequenzen
- Rotieren einer Sequenzfolge im Ring-Speicher
- Mittelwertbildung über mehrere Bilder einer Sequenz
- Sequenz-Binärisierung:
  - Threshold
  - Band (lower and upper threshold)
- Weiterhin können alle 2-dimensionalen Funktionen aus dem picCOLOR-Programm zur Sequenzverarbeitung eingesetzt werden. Diese können über ein MACRO-Programm automatisch oder halbautomatisch ablaufen.
- Object-Tracking in Echtzeit-Bildsequenzen, Stereo-Tracking (mit Sequenz-Tracking-Module)

### 3. Definition eines 3D-Bildes / einer Sequenz

Ein 3D-Bild kann wie eine Sequenz als Folge von 2D-Bildschnitten angesehen werden. Genauso sind derartige Bilder auch im Computerspeicher aufgebaut: man erhält einen Vektor von Zeigern auf 2-dimensionale Bildschnitte, die zusammengenommen ein 3D-Bild oder eine Bildsequenz ergeben. Zur Definition eines 3D-Bildes oder einer Sequenz öffnen Sie die „Define 3D/Sequenz“-Dialogbox im „Control-Panel“. Hier müssen zunächst folgende Größen festgelegt werden:

- Breite der 2D-Bildschnitte (x-Size)
- Höhe der 2D-Bildschnitte (y-Size)
- Tiefe des 3D-Bildes oder Anzahl der Einzelbilder einer Bildsequenz (z-Size)



Diese Parameter können direkt in der oben dargestellten Dialogbox eingestellt werden. Die Angabe der Breite und Höhe kann prinzipiell pixelgenau und damit speichersparend sein. Es empfiehlt sich jedoch trotzdem in vielen Fällen, nicht allzu krumme Zahlen zu wählen, da dies bei der späteren Auswertung eventuell zu umständlicheren Umrechnungen führen könnte. Auch Geschwindigkeitseinbußen beim Übertragen und Kopieren von Bildern sind bei ungeraden Werten möglich. Es hat sich als sinnvoll herausgestellt, wenn die Breite des Speichers durch 8 teilbar ist. Dies ist auch als interner Defaultwert eingestellt, d. h. die Breite wird automatisch soweit vergrößert, daß sie durch 8 teilbar wird. Notwendig wird dies allerdings erst bei der Benutzung bestimmter Bildeinlesekarten, die hardwaremäßig nur bestimmte Breitenstufungen unterstützen. Als maximale Werte für Breite und Höhe sind die Maße der zuvor definierten normalen 2D-Bildspeicher festgelegt. Wenn also 1024\*1024 Pixel große Sequenzen definiert werden sollen, so muß auch der normale 2D-Bildspeicher auf mindestens diese Größe eingestellt werden. Diese Einschränkung wurde vorgegeben, damit es in jedem Falle möglich ist, Schnittbilder in den normalen 2D-Bildspeicher zur Bearbeitung zu übertragen.

Danach kann dann über den „Define Image“-Schalter der Speicherplatz für das mit den obigen Funktionen festgelegte 3D-Bild oder für die Sequenz angefordert werden. Die maximale Anzahl der 3D/Sequenz-Bildspeicher ist begrenzt, z. B. in den normalen picCOLOR-Versionen auf 15 Stück (prinzipiell gibt es natürlich hier kaum eine Grenze – bei Bedarf bitte nachfragen). Per Definitionem haben alle 3D/Sequenz-Bilder die gleiche Größe. Sobald ein 3D/Sequenz-Bild definiert ist, kann die Größe in der Dialogbox nicht mehr verändert werden. Zur Änderung müssen alle 3D/Sequenz-Bilder wieder freigegeben werden.

Nach der Definition verlassen Sie die Dialogbox. Die Sequenz oder das 3D-Bild ist nun im Multiple-Document Hauptfenster des picCOLOR-Programmes als zusätzliches Fenster sichtbar und ist aktiv. Über die üblichen Aktivierungsfunktionen von Windows (entweder Anklicken des Fensters oder Auswahl über das „Windows“-Menü im Hauptfenster) kann die Sequenz oder das 3D-Bild aktiviert oder deaktiviert werden. In der Kopfzeile erscheint der Schriftzug „3D/Sequenz Image X“ mit „X“ als der tatsächlichen Bildnummer, beginnend mit „1“. Man wird feststellen, daß sich bei aktivem Sequenz/3D-Bild auch die Hauptmenü-Leiste ändert: Statt des „File“-Menüs sieht man hier dann das „3D/Sequenz“-Menü. Dies hat mehrere Gründe: erstens wird hiermit die Verwechslungsgefahr mit normalen 2D-Bildern vermindert und zweitens wird verhindert, daß externe Programme anderer Hersteller (wie z. B. die OCR-Software Omnipage) das 3D/Sequence-File-Menü finden und dort eigene Funktionen eintragen, was ja bei 3D/Sequenzbildern kaum sinnvoll wäre. Die Funktionen der Hauptmenüzeile sind komplett geändert, auch wenn einige Funktionen ähnliche Bezeichnungen tragen. Lediglich das „Control-

Panel“ ist gleich. Auch die Funktionen der „Tool-Box“, die mit der rechten Maustaste oder über das Help-Menü aufgerufen werden kann, sind geändert.

Das Freigeben der 3D/Sequenz-Bildspeicher an das System geschieht in umgekehrter Reihenfolge, in der die Speicher angefordert wurden, d. h. man muß das zuletzt angelegte 3D/Sequenz-Bild zuerst schließen. Hierdurch wird sichergestellt, daß intern die Speicher eine fortlaufende Nummerierung besitzen und kein Zugriff auf nichtexistente Speicher versucht wird. Das Schließen kann wie üblich über die Funktionen in der Titelleiste der Fenster erfolgen. Bei Schließen des picCOLOR-Programmes werden automatisch alle noch geöffneten 3D/Sequenz-Bilder geschlossen.

3D/Sequenz-Bilder werden wenn möglich direkt im Hauptspeicher des Computers definiert. Nur so lassen sich Sequenzen auch mit Hochgeschwindigkeitskameras in Echtzeit aufnehmen. Auf die Festplatte ausgelagerte Speicherbereiche wären nicht schnell genug zugreifbar. Damit kann die maximale Sequenzlänge oder 3D-Bildgröße aus dem vorhandenen Speicherausbau bestimmt werden. Hierzu muß man jedoch beachten, daß WINDOWS nicht den kompletten Hauptspeicher zur Verfügung stellt. Es reserviert sich einen recht großen Bereich für allgemeine Systemanforderungen oder zur Ausführung anderer Programme. In der Regel kann man unter WINDOWS NT4.0 etwa 80% des gesamten Hauptspeichers für Sequenzen erhalten, unter WINDOWS 2000 etwas weniger. Bei WINDOWS XP ist die Speicherverwaltung wohl besser geworden. Man erhält zwar auch nur etwa 70% des vorhandenen Speichers als realen Speicher zugeteilt, aber bei größeren Anforderungen funktioniert das Auslagern auf eine „Swap“-Datei recht problemlos. Dadurch wird natürlich jede Bearbeitung deutlich langsamer. Je nach eingesetzter Bildeinlesekarte kann dieser Speicheranforderungswert jedoch sehr viel geringer sein. So verlangen z. B. verschiedene Karten, den Einlesepuffer beim Systemstart (Booten) des Computers zu definieren und festzulegen. Mit der älteren F64P-Karte von CORECO z. B. erhält man damit dann lediglich 20% des Hauptspeichers für Videosequenzen. Ähnlich sieht es bei der Micro-Enable-Karte von Silicon-Software aus, die ebenfalls aus bestimmten Gründen nur relativ kurze Sequenzen zuläßt. Dagegen gibt es bei der VIPER-, der X64- und der Xcelera-Kartenserie der Firma DALSA/CORECO jedoch kaum Beschränkungen: je nach eingesetzter Versionsnummer der SAPERA-Driver-Library kann der gesamte von WINDOWS zur Verfügung gestellte Speicher als Sequenzspeicher benutzt werden. (Bei der alten Version 3.x gibt es lediglich eine Begrenzung auf ungefähr 10.000 Bilder einer Sequenz. Sonst ist in picCOLOR die maximale Sequenzlänge auf 16.000 intern festgelegt). Bei einer normalen 25Hz-Videokamera wären dies mehr als 10 Minuten Video mit 7 GByte benötigtem Speicher, also wesentlich mehr als man unter jedem 32Bit-Betriebssystem ansprechen könnte... Bei der Hochgeschwindigkeitskamera MC1310 erhält man mit diesen 16.000 Bildern bei z. B. 1000 Hz Videofrequenz und einer Bildauflösung von 1280\*512 Pixeln immerhin noch eine Sequenzlänge von 16 Sekunden bei einem Speicherbedarf von allerdings etwa 10 GByte. Bei solchen längeren Sequenzen kann man jedoch auch auf die Streaming-Funktionen von picCOLOR ausweichen, wenn man ein schnelles Festplattensystem hat.

Es ist allerdings zu beachten, daß WINDOWS die benötigte Speichergröße nicht immer freiwillig hergibt. Wenn z. B. größere Programme oder Datenbereiche im Hauptspeicher liegen oder ein großer Teil des Betriebssystems im physikalischen Speicher liegt, dann muß zuerst dafür gesorgt werden, daß der benötigte Speicher freigestellt wird. Schließen Sie hierzu alle nicht benötigten Programme und sorgen Sie dafür, daß auch WINDOWS sich soweit wie möglich auslagert. Letzteres geschieht über bestimmte Tricks. Z. B. kann man sukzessiv immer größere Speicherbereiche anfordern und wieder freigeben. So seltsam dies auch klingt, erst hierdurch akzeptiert WINDOWS (zumindest NT oder 2000), daß man die größeren Speicherbereich auch wirklich benötigt und macht sich selbst sehr klein, d. h. lagert alle nicht benötigten Teile des Betriebssystems auf den Festplatten-Swap-Bereich aus. Natürlich gibt es hierzu auch besser definierte interne WINDOWS-API-Funktionen, die im picCOLOR-Programm leider jedoch nicht angewandt werden können, da die meisten Bildeinlesekarten-Treiber Speicheranforderungsroutinen zur obligatorischen Benutzung zur Verfügung stellen. Wenn Sie also bei der Definition einer sehr langen und großen Videosequenz eine Fehlermeldung erhalten, so versuchen Sie bitte, über die mehrfache Definition oder über die Definition eines weniger zerstückelten Speichers (weniger Bildschnitte, größere x,y-size) WINDOWS von Ihrem Speicherwunsch zu überzeugen. Hierzu ist in der Definitions-Dialogbox der Schalter „Outwit NT“ gegeben, der bewirkt, daß picCOLOR versucht, vor der Definition der einzelnen Bildschnitte einen einzigen zusammenhängenden Speicherbereich der kompletten geforderten Größe zu allocieren und wieder freizugeben. Dieser Trick ist jedoch noch nicht der Weisheit letzter Schluß...

### ROI-Bearbeitung im 3D/Sequenz-Bild

Die von normalen 2D-Bildern bekannten Regions of Interest (ROI) sind für 3D-Bilder oder Sequenzen zumindest in dieser Form nicht bzw. noch nicht definiert und implementiert. Hier wird es als sinnvoller erachtet, das 3D/Sequenzbild in genau der benötigten Größe zu definieren. Dies ist sicherlich wesentlich speichersparender. Trotzdem gibt es natürlich eine Menge Aufgaben, bei denen auch im 3D- oder Sequenz-Bild eine ROI-Bearbeitung sinnvoll oder notwendig ist. In zukünftigen Versionen des picCOLOR-Programmes wird es daher möglich sein, eine 3D-ROI zu definieren, in der dann alle Bearbeitungen stattfinden. Wenn jedoch ein Schnittbild einer Sequenz oder eines 3D-Bildes in einen normalen 2D-Bildspeicher zur weiteren Bearbeitung übertragen wird, so kann hier dann wie üblich eine 2D-ROI gesetzt werden. Die Übertragung selbst erfolgt ebenfalls in die oder aus der aktuellen ROI des 2D-Bildes. Unter bestimmten Bedingungen, z. B. wenn über ein MACRO-Programm mit 2D-Funktionen direkt im 3D/Sequenz-Speicher gearbeitet wird, und außerdem bei bestimmten Spezialfunktionen wird auch jetzt schon eine ROI benötigt. In diesem Fall wird dann die allgemeine im 2D-Bild definierte gerade aktuelle ROI benutzt. Hierbei sollte beachtet werden, daß ein Sequenzbild definitionsgemäß in x/y-Ausdehnung auch kleiner als ein 2D-Bild sein kann. Eine zu große ROI wird dann auf die zulässige Größe reduziert.

## 4. Darstellung von 3D-Bildern/Sequenzen

Zur Darstellung einer Sequenz oder eines 3D-Bildes gibt es verschiedene Möglichkeiten: in einer Video-Player-Dialogbox kann die Sequenz mit unterschiedlichen Optionen abgespielt werden, einmal vorwärts, oder ständig wiederholend in der Art eines Ringspeichers, mit einem ausgewählten Bereich, unterschiedlichen Geschwindigkeiten usw. Dies geschieht direkt bei aktivem 3D/Sequenz-Bildspeicher.

3D-Processing -Active Buf. -> -Destin.Buf. -> Pick 3D-Image ROI -> 3D (xy) 3D (xy) -> ROI 3D (zy) -> ROI 3D (xz) -> ROI	Sequence Processing - Active Seq. -> - Destin Seq. -> Pick Seq.-Image ROI -> Sequence (xy) Seq. (xy) -> ROI Seq. (xt) -> ROI Seq. (yt) -> ROI
--	--

*3D- und Sequenz-Menü vom 2D-Bildspeicher aus gesehen, im Processing-Menü*

Weiterhin ist die Möglichkeit gegeben, alle Bilder der Sequenz (oder alle Schnitte des 3D-Bildes) gleichzeitig auf dem Bildschirm darzustellen und mit der Maus eines davon auszuwählen und in einen normalen 2D-Bildspeicher zu übertragen (Pick 3D-Image). Schließlich kann jedes einzelne Bild aus beliebiger (orthogonaler) Ansicht (Sequenz: xy, xt, yt, oder für 3D-Bilder: xy, yz, zx) anhand seiner Bildnummer aus dem Speicher auf einen 2D-Bildspeicher geholt werden. Letzteres wird bei aktivem 2D-Bildspeicher über die oben dargestellten Funktionen im Processing-Menü durchgeführt.

Wenn mehr als ein 3D-Bild oder mehrere Sequenzen definiert wurden, so kann vor der Übertragung mit der Funktion "Set Active 3D-Buffer" oder "Set Active Sequence" das angesprochene 3D/Sequenz-Bild ausgewählt werden.

Eine einfache Pixelinspektionsfunktion für xyz-(oder xyt)-Koordinaten und Grauwert ist im normalen Anzeigemodus des Bildspeichers möglich: klickt man an einer beliebigen Stelle im Bild mit der linken Maustaste, so erhält man ein Fadenkreuz und eine Anzeige der aktuellen Koordinaten des Cursors incl. der Schnittbildnummer der Sequenz oder des 3D-Bildes mit dem dort befindlichen Grauwert in der Statuszeile am unteren Bildrand.

### 4.1. Einzelbilddarstellung

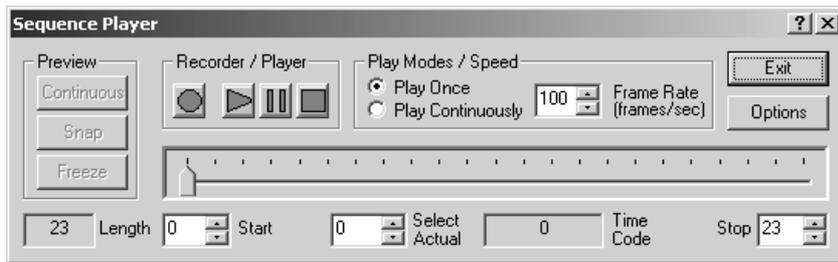
Von einem beliebigen normalen 2D-Bildspeicher aus kann mit den "Seq. (xy) -> ROI" und "3D (xy) -> ROI"-Funktionen jedes Einzelbild einer Sequenz oder eines 3D-Bildes auf den Bildschirm (d. h. den aktuellen Bildspeicher) in die aktive ROI kopiert werden. Einzelbild-Darstellung einer Sequenz ist auch für Orts/Zeit-Bildschnitte möglich, in denen dann z. B. eine Bewegung in x-Richtung oder in y-Richtung in einem einzigen Bild dargestellt werden kann. Die zugehörigen Funktionen sind "Seq.(xt) -> ROI" und "Seq. (yt) -> ROI". Entsprechend dazu gibt es für die Einzelbilddarstellung von 3D-Bildern die Funktionen "3D (xy)->ROI", "3D (yz)->ROI" und "3D (zx)->ROI", mit denen die Anzeige von xy-, yz- oder zx-Schnitten vorgenommen werden kann. Im aktuellen 2D-Bildspeicher in der aktuellen ROI stehen die Bildschnitte dann zur 2-dimensionalen Weiterverarbeitung zur Verfügung. Zusammen mit den geometrischen Spiegelungsfunktionen aus dem "Geometry"-Menü können hiermit dann 3D-Bild-Schnitte von allen Seiten betrachtet werden.

### 4.2. Auswahl aus verkleinerter Darstellung aller Bilder

Da es oft schwierig ist, aus einer langen Sequenz das richtige Bild herauszusuchen, um dieses dann einer weiteren Bearbeitung zu unterziehen, wird mit der "Pick 3D Image to ROI" oder "Pick Seq.-Image to ROI"-Funktion ein einfaches Hilfsmittel gegeben, mit dem alle oder ein bestimmter Teil der Sequenzbilder oder 3D-Schnitte gleichzeitig verkleinert auf dem Bildschirm dargestellt werden können. (Hierzu kann die Bildnummerschrittweite (Interleaving) eingegeben werden, d.h. mit 1: jedes Bild wird gezeigt; mit 2: jedes zweite Bild wird gezeigt usw.; diese Möglichkeit ist z. Zt. jedoch nicht implementiert). Dann kann mit der linken Maustaste eines der dargestellten Bilder ausgewählt werden. Dieses wird dann in die aktuelle ROI kopiert und kann dort direkt weiterverarbeitet werden. Die rechte Maustaste oder ESC bricht die Funktion ohne Bildspeicheränderung ab.

### 4.3. Videoplayer für 3D-Bild und Sequenz

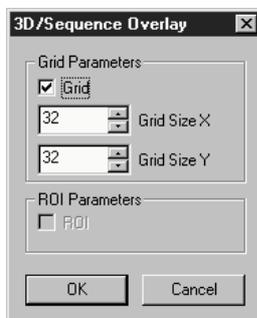
Mit einem Software-Videoplayer können Sequenzen oder 3D-Bilder wie mit einem realen Videorekorder abgespielt und betrachtet werden.



- Play, Pause, Stop: Steuerung der Videoanzeige. Pause und Stop sind nur im Play-Continuously-Modus nutzbar.
- Play Once/Continuously: Die Sequenz wird nur einmal oder fortlaufend wiederholt dargestellt.
- Frame Rate: Geschwindigkeit des Sequenzablaufes in Bildern/Sekunde: Bei einem Eingabewert von ,1‘ bis ,100‘ wird die Sequenz mit etwa dieser eingestellten Frame Rate dargestellt, allerdings nicht sehr genau, da intern der 16ms WINDOWS-Timer benutzt wird. Bei Werten größer als ,100‘ wird die Sequenz so schnell wie möglich auf dem Bildschirm ohne Benutzung eines Timers abgespielt. Dies kann speziell bei kleineren xy-Ausdehnungen sehr schnell sein, wesentlich schneller als Videoechtzeit.
- [Start..Stop]: Es kann eingestellt werden, daß nur ein Teil der Sequenz fortlaufend abgespielt wird. Hierzu müssen die Bildnummern des ersten und des letzten Bildes der Teilsequenz eingegeben werden.
- SelectActual: Zur Einzelbilddarstellung kann entweder die Schnittbildnummer mit den up/down-Schaltern eingegeben werden oder es kann auch der Bild-Schieber benutzt werden.
- TimeCode ist der Zeitstempel eines Bildes, der entweder vom Frame Grabber oder vom High-Resolution-Timer (unter WINDOWS) vergeben wurde.

Über den „Options“-Taster erreicht man noch einige Einstellungen, die jedoch z. Zt. nur die Bildübernahme betreffen. Wenn das Stereo-Flag gesetzt ist und die richtigen Sequenzspeicher definiert sind, dann wird das System beide Stereo-Bildsequenzen gleichzeitig anzeigen, allerdings nicht mit dem Bildaufbau auf dem Monitor synchronisiert. Falls dies gewünscht ist, z. B. für einen Display für eine Stereobrille (Shutter-Brille), so wird dies über spezielle Funktionen des Stereo-Extension-Modules durchgeführt (DirectX), die auch Zugriff auf die Timings der Austastlücken haben.

#### 4.4. Graphische Overlay-Funktionen für die Videodarstellung



Über die „Overlay“-Funktion im „Graphics“-Menü kann eine Art Overlay-Graphik definiert werden, die beim Abspielen eines Videos über das jeweilige aktuelle Sequenzbild gelegt wird. In der Regel ist dies ein Gitter bestimmter Maschenweite, daß z. B. bei der Beurteilung von Partikelverteilungen sehr hilfreich sein kann.

#### 4.5. Ausgabe von Sequenzen auf einem Videorekorder

Oft ist es erwünscht, eine aufgenommene und bearbeitete Videosequenz auf einen Videorekorder zu überspielen. Dies ist mit zusätzlicher Hardware möglich, die von vielen modernen Graphikkarten mit einem TV-Output zur Verfügung gestellt wird. Hierzu muß dann lediglich die Bildgröße an die gewünschte Videoauflösung angepaßt werden, das Video sinnvoll auf dem Bildschirm platziert werden, z. B. bildfüllend, und es muß der Video-Ausgang der speziellen Graphikkarte aktiviert werden. PicCOLOR nutzt nicht die Windows-Interne Möglichkeit, daß Videos automatisch bildfüllend auf den TV-Ausgang gelegt werden.

#### 4.6. Ausgabe als AVI-Video-File (siehe auch Kap. 6, Laden und Speichern)

Sequenzen können direkt in das Videoformat AVI umgewandelt werden. In der implementierten Version kann z. Zt. allerdings noch keine Kompression eingegeben werden. Da AVI-Sequenzen im RGB-Format gespeichert werden, wird also der Speicherbedarf von Grauwertbildern verdreifacht. Mit anderen externen Programmen, z. B. VirtualDub, kann das Video dann mit allen verfügbaren Video-Codecs komprimiert werden.

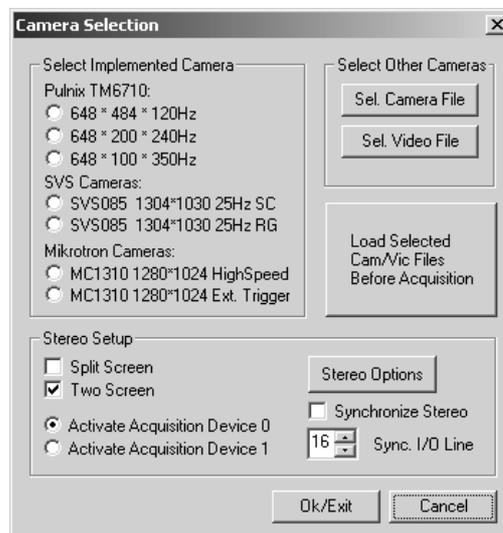
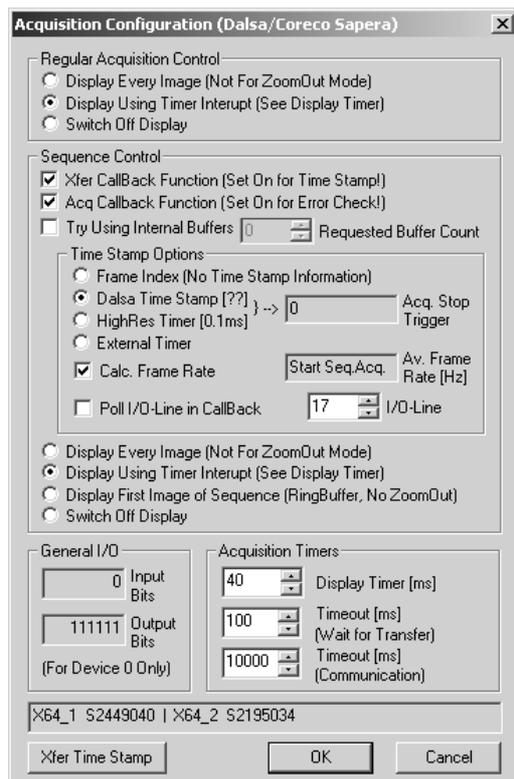
## 5. Sequenz-Bildübernahme

Bei der Bildübernahme einer Sequenz soll in der Regel Videoechtzeit erzielt werden, d. h. die Bilder der Sequenz sollen genau mit der Geschwindigkeit in den Sequenzspeicher übertragen werden, mit der sie von der Kamera geliefert werden. Selbstverständlich sind die meisten modernen FrameGrabber hardwaremäßig darauf eingerichtet, die Bilder immer in Echtzeit entweder in ihren eigenen „Onboard“-Bildspeicher (Video-RAM) oder direkt in den Hauptspeicher des Computers einlesen zu können. Bei Videovollbildsequenzen mit z. B. 768\*576 Pixeln und 25 Bildern/Sekunde (bei CCIR) beträgt der Videostrom üblicherweise 14.75 MByte pro Sekunde. Wegen der recht großen Austastlücken zwischen zwei Bildern (Vertikalaustastlücke) und auch zwischen zwei Zeilen (Horizontalaustastlücke) ist die tatsächlich benötigte gemittelte Videobandbreite jedoch "nur" etwa 10 MByte/Sekunde. Bei Hochgeschwindigkeitskameras kann jedoch leicht ein wesentlich höherer Wert erreicht werden, z. B. bis maximal 850MB/s über die CameraLink-Schnittstelle. In naher Zukunft werden allerdings über die HS-Link-Schnittstelle (DALSA) noch deutlich höhere Übertragungsgeschwindigkeiten erzielt. Um solche Datenmengen in den Speicher einlesen zu können, wird das Verfahren des Direct Memory Access (DMA) benutzt. Um auch längere Videosequenzen aufnehmen zu können, wird in der Regel eine Technik mit der Bezeichnung „Scatter Gather“ benutzt, die dafür sorgt, daß die kleinen Speicherblöcke, die von WINDOWS z. B. mit der Größe von 4kByte zur Verfügung gestellt werden, in der notwendigen Geschwindigkeit in den physikalischen Speicherbereich zu liegen kommen und mit DMA beschrieben werden können. Dieser Vorgang benötigt praktisch keinerlei CPU-Zeit. Die in den Hauptspeicher geladenen Bilder können dann entweder in Echtzeit oder im Post-Processing ausgewertet werden oder auf Festplatte gespeichert werden. Mit den heutigen schnellen Festplatten ist es unter Umständen auch möglich, längere Videosequenzen direkt auf die Platte zu schreiben. Mit recht einfachen RAID0-Plattensystemen (SATAII oder SCSI) wurden bis zu 300MB/s erreicht. Darüber hinaus wird es allerdings komplizierter und teuer. Sehr interessant werden hier sicher bald die SSD-Plattensysteme, besonders wenn sie als PCIe-Karte direkt und mit hoher Speicherbandbreite ansprechbar sind. Von Fusion-IO sind Systeme verfügbar, die bis zu 1500MB/s speichern können sollen, und das bei immerhin bis zu 2TB Speicherplatz! Hierzu stehen die Tests allerdings noch aus.

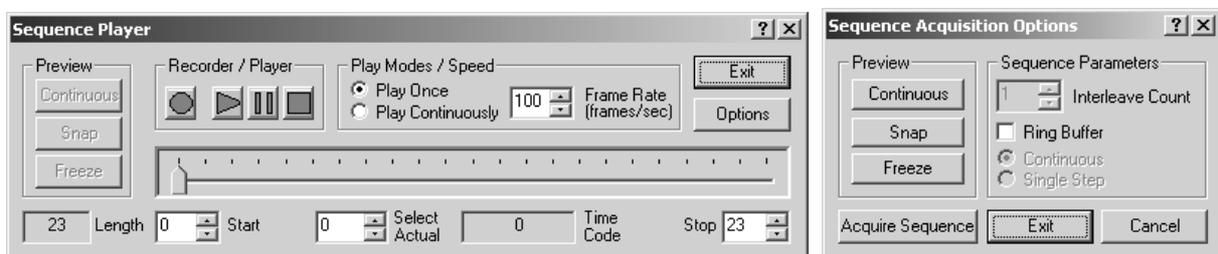
Zur normalen Sequenzaufnahme öffnet sich eine Video-Recorder-Dialogbox, die dem Video-Player sehr ähnlich ist. Es sind lediglich einige weitere Optionen aufrufbar und der Aufnahme (Record)-Taster ist hier natürlich wirksam.

### 5.1. Parameter für die Sequenzaufnahme

(Achtung: z. Zt. sind nicht alle hier beschriebenen Funktionen implementiert. Einige der hier beschriebenen Funktionen stammen aus dem ursprünglichen picCOLOR V3.0 Programm, das aufgrund der eingesetzten proprietären Hardware spezielle Möglichkeiten hatte, Kameras sehr hardware-nah anzusprechen)



Die Configure-Acquisition-Dialogbox und die Select-Camera-Dialogbox finden Sie im Acquisition-Menü. Dazu gibt es noch die bereits erwähnte Option-Box des Video-Recorders. In diesen Dialog-Boxen lassen sich eine Reihe von Parametern für die Sequenzaufnahme einstellen, z. B. die Ringspeichereinstellung, der Einzelbild-Modus oder die kontinuierliche Aufnahme, die Split-Screen- oder Stereo-Übernahme durch zwei Kameras, die Art der Bilddarstellung während der Aufnahme, die Art der Callback-Funktion zur weiteren Behandlung der eingelesenen Bilder, verschiedener Zeitstempelmöglichkeiten, und weitere Parameter. Die meisten dieser Parameter sind die gleichen wie bei der normalen 2D-Bildaufnahme und sind im Haupthandbuch erklärt. Als Besonderheit kann hier etwa die Bilddarstellung während der Aufnahme angesehen werden, die in der Configure-Acquisition-Dialogbox eingestellt werden kann. Besonders bei Hochgeschwindigkeitsaufnahmen, z. B. mit 1000 Bildern/s, ist es schließlich nicht sinnvoll, alle Bilder auf dem Bildschirm darzustellen. Das kann kein Bildschirm, man würde es auch nicht erkennen können, und es würde viel zu viel Rechenzeit verbrauchen. Übliche Einstellung ist hier die Darstellung über einen Timer-Interrupt, z. B. mit 25 Hz. Dieser „DisplayTimer“ ist der gleiche wie bei der Einstellung für reguläre Bilder. Bei Ring-Speicher-Einstellung kann es auch sinnvoll sein, nur jedes erste Bild des Speichers anzuzeigen. Wichtig ist hier auch die Möglichkeit, eine oder mehrere „CallBack“-Funktionen einstellen zu können. Der Frame Grabber Treiber ruft nach dem Übertragen eines Bildes in den Hauptspeicher dann eine spezielle Funktion auf, die z. B. den Zeitstempel ausliest, die auch Fehler prüfen kann, und die etwa einer anderen Echtzeit-Funktion mitteilen kann, daß das neue Bild zur Bearbeitung freigegeben ist.



#### Parameter in der Sequenz-Player/Recorder-Dialog-Box:

Für die Sequenzaufnahme wird eine Video-Recorder Dialogbox geöffnet, die der Video-Player Dialogbox sehr ähnlich ist. Hierzu muß natürlich mindestens ein Sequenzspeicher definiert sein und es müssen zuvor die notwendigen Sequenz-Kontrollparameter eingestellt sein. Wenn mehrere Sequenzen definiert wurden, so muß vor der Sequenzaufnahme der Speicher ausgewählt, d. h. aktiviert werden, in den dann die Sequenz eingelesen wird. Bei Stereo-Aufnahmen ist es egal, welche der beiden Stereo-Sequenzen beim Aufruf aktiv ist: das System kennt die beiden Stereo-Sequenzen und aktiviert diese beim Aufruf des Sequenz-Rekorders..

#### Record-Button (Aufnahme-Schalter)

Der Schalter "Record" startet die Sequenzaufnahme. Die Sequenzaufnahme wird automatisch nach Einlesen des kompletten Sequenzspeichers beendet, wenn der Speicher nicht als Ringspeicher definiert ist (in der „Option“-Box). Sie kann jedoch bei den meisten Frame Grabbern mit der ESC-Taste vom Benutzer abgebrochen werden. Bei Ringbuffer-Einstellung wird kontinuierlich immer weiter eingelesen - es muß dann schließlich ein Abbruch mit der ESC-Taste (oder auch mit der rechten Maustaste) erfolgen. Über das Echtzeit-Zusatzmodul (Real Time Extension Module) ist es zusätzlich möglich, die Sequenzaufnahme über spezielle hardware- oder softwaregetriggerte Signale zu starten und zu beenden. Die Sequenzaufnahme wird immer bei Bild 0 begonnen. Im Single-Step-Modus kann man mit dem Record-Button auch die Aufnahme in Einzelbildern steuern. Jeder Mausklick auf den Record-Button liest ein weiteres Bild ein. Dies kann z. B. für Standbilder vom Videorekorder oder für Einzelschnitte eines 3D-Bildes vom Mikroskop durch Durchfokussieren interessant sein. Der Schiebepfeil ist dabei aktiv und kann beliebig eingestellt werden, d. h. man kann einstellen, ab welchem Bild die Sequenz übernommen wird. Dabei wird jeweils in den nächsten Bildspeicher geschrieben, d. h. wenn der Schieber auf Bild N steht, wird beim nächsten Record-Button-Klick ein Bild in den Speicher N+1 übernommen. Wenn das letzte Bild im Sequenzspeicher übernommen wurde, wird eine Meldung ausgegeben und die Funktion beginnt wieder bei Bild 0. Hierbei muß außerdem noch beachtet werden, daß bei bestimmten FrameGrabber-Karten nicht direkt in beliebige Bildspeicher einer Sequenz übernommen werden kann sondern lediglich in das erste Bild einer Sequenz. Die picCOLOR-Software benutzt daher in allen Fällen das Bild 0 des Sequenzspeichers als Zwischenspeicher und kopiert das eingelesene Bild dann auf das gewünschte Bild im Sequenzspeicher. Beim bearbeiten oder abspeichern der Sequenz muß also beachtet werden, daß Bild 0 immer ein ungültiges Bild ist, wenn per Single-Step übernommen wurde.

**Interne/externe Synchronisation:** Wenn nur eine Kamera eingesetzt wird, hat man die Möglichkeit, sie frei durch ihre eigene Zeitbasis getriggert laufen zu lassen („free-running“ Modus) oder sie auf eine externe Triggerfrequenz zu synchronisieren. Bei mehr als einer Kamera, also speziell im Stereo-Modus mit zwei Kameras, müssen natürlich beide exakt synchron laufen. Hierzu muß ein externer Takt bereitgestellt werden, der entweder direkt zu den Kameras geführt wird oder durch die Frame Grabber hindurch zu den Kameras geleitet wird. Das bedeutet, daß hier in beinahe allen Fällen der Frame Grabber extern, d. h. von außen getriggert wird. Bei modernen Systemen mit digitalen Kameras und Frame Grabbern wird es hier kaum zu Problemen kommen: der Frame Grabber wird sich immer problemlos darauf einstellen und alle ankommenden Bilder einlesen. Sogar wenn das Triggersignal kurzzeitig abgeschaltet wird und keine Bilder mehr von der Kamera ankommen, wird der Grabber geduldig warten. Auf diese Weise können komplizierte Messungen mit exakten Start- und Stop-Zeiten hardwaremäßig von außen eingestellt werden.

Bei älteren und vor allem bei analogen Frame Grabbern ist dies allerdings nicht so selbstverständlich. Am einfachsten wäre es noch, wenn der Frame Grabber selbst dieses Synchronsignal zur Verfügung stellt und sich daher nicht darauf einstellen muß. Speziell beim Einsatz mehrerer Kameras ist die Synchronisation über die karteneigene Zeitbasis dann die optimale Methode: alle Kameras werden gemeinsam von der Bildeinlesekarte synchronisiert - es kann zu praktisch beliebigen Zeiten zwischen verschiedenen Kameras umgeschaltet werden, im Extremfall auch innerhalb der gerade übernommenen Zeile, wodurch sehr spezielle Effekte und Aufnahmesituationen möglich werden wie z. B. eine zeilen- oder sogar spaltenweise Teilbildaufnahme mehrerer Kameras. Zur externen Synchronisation des Frame Grabbers dagegen durch die Kamera oder eine andere Videoquelle müssen die Synchronisationssignale bestimmte Mindestanforderungen an Gleichmäßigkeit und Regelmäßigkeit erfüllen, da hierzu in der Regel PLL-Schaltkreise verwendet werden. Eine längere Auszeit wie beim digitalen Frame Grabber ist hierbei absolut nicht möglich. Diese notwendigen exakten Synchronsignale liegen z. B. auch bei Videorekordern oft nicht vor, so daß man hier keine allzu guten Ergebnisse erwarten sollte. Wenn z. B. beim Videosignal eines Videorekorders einzelne Synchronisationspulse fehlen oder nicht erkennbar sind, so wird die FrameGrabber-Hardware versuchen, über eine PLL-Steuerung selbst welche einzufügen. Damit wird er immerhin in der Lage sein, das Bild einzulesen, es kann jedoch sein, daß hierdurch Bildsprünge oder -versätze auftreten.

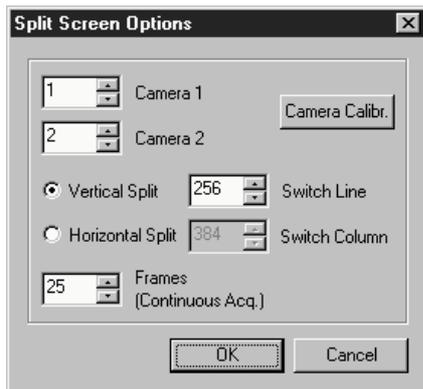
**Interleave Count / Delay Factor:** Einstellung eines Verzögerungsfaktors für langsamere Aufnahme von Echtzeit bis herunter zu 1 Bild/Stunde. Hierzu wird in einer Eingabebox der Delay-Factor N mit:  $\text{Aufnahmefrequenz} = \text{Normfrequenz} * 1/N$  eingegeben. Bei CCIR-Norm wäre dies also:  $25\text{Hz} * 1/N$ . Maximale Aufnahmefrequenz erhält man beim Wert 1: es wird dann jedes ankommende Bild eingelesen. Beim Wert 2 wird jedes zweite Bild eingelesen, man arbeitet also mit halber Frequenz und erhält nur noch 12.5 Vollbilder pro Sekunde. Der maximale Wert von 90000 ergibt eine Bildfrequenz von 1 Bild pro Stunde. Diese Einstellung ist nur für bestimmte Frame Grabber möglich. Wenn diese Funktion nicht implementiert ist, z. B. bei allen digitalen Karten, so kann in der Regel die genaue Bildfrequenz durch eine direkte Ansteuerung der Kamera geschehen. So ist bei einigen analogen Kameras eine „Field-On-Demand“ oder „Restart/Reset“-Steuerung möglich. Bei digitalen Kameras kann in der Regel entweder beim freilaufenden Modus über eine Ansteuerungs-Software die genaue Bild-Rate eingestellt werden, oder im extern getriggerten Modus (asynchrone Triggerung) über die Ansteuerung mit einer speziellen Triggerfrequenz. Die Frame Grabber Firmware ist flexibel genug, auf die einzelnen Bilder der Kamera zu warten, auch bei sehr niedrigen Frequenzen. Hierzu müssen je nach Frame Grabber Fabrikat eventuell speziell angepaßte Kamera-Definitionsfiles erstellt werden, die dann von Grabber-Treiber eingelesen werden.

**Ring Buffer:** Während der Bildaufnahme einer Videosequenz kann der Sequenz-Speicher als Ring-Speicher benutzt werden. Wie üblich wird dann die Bildaufnahme der Sequenz mit der linken Maustaste oder einer anderen Triggerung (siehe Echtzeit-Zusatzmodul) gestartet. Beendet wird die Bildaufnahme mit einem weiteren Drücken der linken oder der rechten Maustaste oder der ESC-Taste oder einer weiteren zuvor definierten Triggerung. Die Bilder werden kontinuierlich in den Sequenzspeicher eingelesen. Bei Überschreiten des Sequenzspeicherendes wird wieder am Anfang begonnen und alte Teile der Sequenz werden überschrieben. Bei Abbruch der Übernahme ist der Anfang der Sequenz dann natürlich nicht notwendigerweise am Anfang des Sequenz-Speichers. Mit der Funktion "Rotate" (im Sequence-Processing-Menü) kann bei Eingabe von "-1" (=automatic) der gesamte Ring-Speicher so gedreht werden, daß der Anfang der Sequenz zum Beginn des Speichers geschoben wird. (Z. Zt. wird tatsächlich intern geschoben, es werden nicht einfach nur Pointer geändert, daher kann die Funktion etwas länger dauern). Mit dieser Ring-Speicher-Funktion können einmalig auftretende Vorgänge leicht eingelesen werden, indem man einfach die Bildübernahme stoppt, sobald dieser gewünschte Vorgang eingetreten ist. Je nach Länge des Ringspeichers hat man dann eine Sequenz von z. B. mehreren Sekunden vor dem Vorgang bis zu mehreren Sekunden nach dem Vorgang.

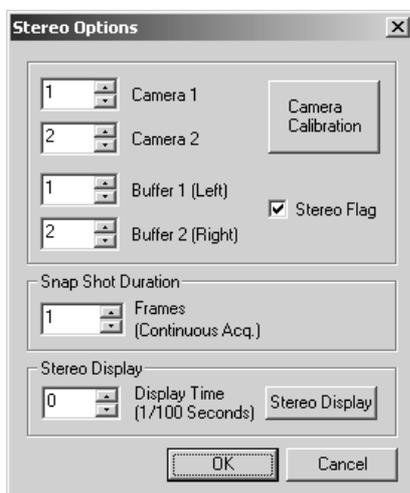
**Continuous/Freeze/Snap Buttons in der Acquisiton Dialogbox:** Diese Funktionen sind z. Zt. im picCOLOR-Programm folgendermaßen definiert: Mit dem Continuous-Schalter wird eine kontinuierliche Bildübernahme gestartet, die jedoch nur das erste Schnittbild der Sequenz überschreibt. Diese Funktion dient lediglich zur live-

Ansicht des Bildes, z. B. zum Fokussieren der Kamera. Das gleiche gilt für den Snap-Shot-Schalter: es wird ein einziges Bild in das erste Schnittbild der Sequenz übertragen. Mit Freeze wird die zuvor gestartete live-Bildübernahme gestoppt. Achtung: Diese Funktionen werden in zukünftigen Programmversionen möglicherweise geändert!

**Interlaced / Non-Interlaced:** CCIR oder EIA-(RS-170)-Norm-Kameras arbeiten üblicherweise im Interlaced-Modus. Wenn eine solche Kamera von der Frame Grabber Karte synchronisiert wird, so kann dies im üblichen Interlaced-Modus, aber auch im Non-Interlaced-Modus geschehen. Wenn es sich in der Aufnahme um schnelle Bewegungen handelt und die vertikale Auflösung eine nicht allzu große Rolle spielt, so empfiehlt es sich, auf Non-Interlaced zu stellen. Hierdurch erhält man die doppelte Aufnahmefrequenz (z. B. 50 Hz bei CCIR oder 60 Hz bei RS-170) bei halber y-Auflösung. Hierzu muß die Kamera natürlich extern synchronisierbar sein und noninterlaced Signale erkennen und umsetzen können.



**Split-Screen-(Teilbild-)Bildübernahmen:** Das Bildeinlesen von Teilbildern von mehreren Kameras ist mit verschiedenen Frame Grabbern möglich. Der ursprüngliche Sinn der Split-Screen-Bildübernahme war die Möglichkeit, mit einfachen analogen Kameras und einem einzigen Frame Grabber mit gemultiplexten Eingängen Stereo-Bilder aufnehmen zu können. Hier wurde dann z. B. nach der Aufnahme eines Halbbildes auf die zweite Kamera umgeschaltet. So hatte man dann beide Stereo-Bilder, wenn auch nur jeweils ein Halbbild, im Bildspeicher und konnte die Auswertung durchführen. Auch mehr als zwei Kameras konnten so eingesetzt werden. Heute mit all den flexiblen digitalen Frame Grabbern wird eine solche Anwendung wohl kaum noch interessant sein, aber in Spezialfällen kann dieses Verfahren dennoch sinnvoll sein. Ein interessantes Beispiel wäre etwa eine extrem teure Höchstgeschwindigkeitskamera, etwa mit 1 Million Bildern pro Sekunde, von der wahrscheinlich nur eine einzige zur Verfügung steht. Hier könnte man mit einer speziell ausgerichteten Spiegelanordnung (auch auf dem Markt erhältlich) Stereo-Teil-Bilder aus zwei verschiedenen Ansichten erzeugen. Für eine 3D-Tracking-Aufgabe muß dem picCOLOR-System dann mitgeteilt werden, wo die Trennungslinie zwischen den beiden Teilbildern liegt. Die picCOLOR-Software ist so ausgelegt, daß Teilbilder sowohl nebeneinander als auch übereinander angeordnet sein können: **Switch Line:** Die Bildzeile, bei der von Kamera 1 auf Kamera 2 umgeschaltet wird, kann in weiten Grenzen eingestellt werden. Bei Interlaced-Kameras muß hierbei die Halbbild-Zeile eingestellt werden, an der umgeschaltet werden soll. **Switch Column:** Die Bildspalte, bei der von Kamera 1 auf Kamera 2 umgeschaltet wird. Die Variable „Frames (continuous acq.)“ wird z. Zt. nicht benutzt.



**Stereo-Bildübernahme:** Hierbei werden die Bilder zweier Kameras in zwei getrennte Bildspeicher eingelesen. Dies können normale Bildspeicher, aber auch Bildsequenzen sein. Achtung: damit die Stereo-Aufnahme starten kann, müssen die Bildspeicher für linke und rechte Kamera unterschiedlich gesetzt sein, es muß das allgemeine Stereo-Flag gesetzt sein, und es muß zusätzlich dem Frame Grabber gesagt werden, daß eine Stereo-Aufnahme erfolgen soll. Letzteres geschieht über die Split-Screen und Two-Screen Schalter in der Camera Selection Dialogbox. Wenn zwischen zwei Aufnahmen auf eine andere Bildspeicher-Art umgeschaltet wird, d. h. etwa von Sequenzbildern auf normale 2D-Bilder, dann wird intern das Stereo-Flag überprüft und entsprechend angepaßt, wenn z. B. zu den eingestellten Bildspeichernummern keine entsprechenden Bildspeicher existieren. Die Variable „Frames (continuous acq.)“ wird z. Zt. nicht benutzt. Mit der „Display Time“ kann die Dauer einer Stereo-Sequenz-Anzeige eingestellt werden, die dann mit der „Stereo Display“-Taste per DirectX inclusive einer Ansteuerung für eine LCD-Shutter-Brille für eine richtige 3D-Darstellung gezeigt wird.

## 5.2. Beispiel einer Sequenzaufnahme

Im folgenden Beispiel wird die normale Vorgehensweise zur Aufnahme einer Echtzeitsequenz vorgestellt:

- Definition des 3D/Sequenz-Speichers mit der erforderlichen x,y-Größe zur Aufnahme der Kamerabilder und der gewünschten Länge der Videosequenz.
- Nach der Definition geht man in das „Acquisition“-Menü und öffnet dort die „Camera Selection“-Dialogbox. Hier kann die angeschlossene Kamera ausgewählt werden und es können eventuell bestimmte Kameraparameter gewählt werden. Hier geht es nur um die Mitteilung an den Frame Grabber, welche Kamera angeschlossen ist. Das picCOLOR-Programm ist jedoch normalerweise nicht dafür ausgelegt, eine Kamera direkt zu programmieren. Hierzu werden in der Regel von den Kameraherstellern eigene Programme geliefert, die eventuell vor der Bildübernahme gestartet werden müssen. Hierzu gibt es ein Beispiel im speziellen Handbuch RTPOS\_Usage.pdf. In Zukunft könnte allerdings über die „Gen-I-Cam“-Schnittstelle auch eine generische Schnittstelle zur Verfügung gestellt werden, mit der hersteller-unabhängig alle Kameras eingestellt werden können. Bei der Auswahl der Kamera wird in der Regel ein sogenannter „Kameradefinitionsfile“ ausgewählt und geöffnet. Hierin stehen in einer vom Frame Grabber Hersteller vorgegebenen Form verschiedene Parameter zur Einstellung des Grabbers.
- Eventuell kann es sinnvoll sein, noch die „Acquisition Configuration“-Dialogbox zu öffnen. Hier werden spezielle Einstellungen vorgenommen, die z. B. die Art eines eventuell eingeschalteten Zeitstempels und die Art der Callback-Funktion betreffen. Die Callback-Funktion ist eine interne picCOLOR-Funktion, die automatisch nach jedem eingelesenen Bild aufgerufen wird. Sie kann dann prüfen, ob das Bild in Ordnung ist, rechtzeitig ankam oder angenommen wurde, und ob während der Aufnahme irgendwelche Fehler aufgetreten sind. Sie ermittelt auch den Zeitstempel, falls dieser von einer speziellen Quelle gelesen werden soll, und gibt ein Signal ab, damit andere Echtzeit-Funktionen das neue Bild auswerten können.
- Schließlich kann man die „Frame Grabber“-Dialogbox, d. h. die „Videorecorder“-Dialogbox öffnen, wo die Sequenzaufnahme mit dem „Record“-Taster gestartet werden kann. Zuvor kann man mit dem Schalter „Continuous“ eine normale Bildübernahme starten, die nur in das erste Bild der Sequenz einliest und die als Live-Video sichtbar ist, so daß man z. B. noch die Fokussierung und Einstellung der Kamera überprüfen kann. Wenn bei Betätigung der „Record“-Taste gerade ein Live-Bild eingeschaltet ist, so wird diese Funktion zuerst gestoppt. Hierbei kann ein kleiner Zeitverzug von wenigen Millisekunden entstehen, da die Frame-Grabber-Treiber in der Regel bestimmte Synchronisationszeiten nach Beendigung einer Bildaufnahme benötigen. Hier in der Dialogbox oder ihrer untergeordneten „Option“-Box können weitere Parameter eingestellt werden. Dies kann z. B. die gewünschte Videoquelle bei mehreren Kameraeingängen sein oder der Interleave-(Delay-) Faktor oder die Einstellung des Sequenzspeichers als einfacher Speicher mit Anfang und Ende oder als Ring-Speicher, der immer wieder neu zyklisch beschrieben wird. Die Ring-Speicher-Einstellung ist normalerweise die Defaulteinstellung, da hierbei deutlich mehr Möglichkeiten der Triggerung und Fehlersuche und -ausgabe bestehen.
- Je nach Einstellung in der „Acquisition Configuration“-Dialogbox werden während der Sequenzaufnahme entweder alle eingelesenen Bilder, oder nur – per Timer gesteuert – in bestimmten Intervallen, oder aber gar kein Bild angezeigt. Nach beendeter Sequenzaufnahme kann die Sequenz im Vider-Recorder/Player direkt betrachtet werden. Alle Betrachtungsfunktionen der Dialogbox entsprechen den Funktionen des Video-Players.

Einschub: Arbeiten mit älteren Interlaced-Kameras oder Video-Rekorder: Interlaced Vollbild oder Non-Interlaced Halbbild? Sehr wichtig ist bei der Sequenzaufnahme mit üblichen Kameras die Möglichkeit, die Halbbilder als eigenständige Bilder anzusehen, natürlich auf den Faktor 1/2 in vertikaler Richtung gestaucht: man bedenke, daß es sich hierbei ja üblicherweise um Bewegungssequenzen handelt. Bei 25Hz CCIR-Videofrequenz hat das ODD-Feld den gleichen zeitlichen Abstand von 20 ms zum vorigen EVEN-Feld wie zum nachfolgenden EVEN-Feld. ODD- und EVEN-Felder passen nicht mehr zusammen - eine schnelle Bewegung von Objekten wäre kaum auswertbar und nur als unangenehmes Verwischen des Bildinhaltes zu erkennen, wenn beide Halbbilder gleichzeitig dargestellt oder bearbeitet würden. Eine Zeilenverdopplung oder Interpolation würde hier zwar helfen, jedoch zu unnötigem Datenverlust führen. Viel sinnvoller ist es hier, die einzelnen Halbbilder als

vollwertige Einzelbilder bei halber vertikaler Auflösung einer doppelt so schnellen Sequenz (nämlich mit 50 Bildern/Sekunde bei CCIR) zu betrachten und auch so auszuwerten. Die vertikale Stauchung muß hierbei natürlich durch einen speziellen Kalibrierfaktor berücksichtigt werden. Natürlich kann man auch vor der Bearbeitung vertikal um den Faktor 2 zoomen, was jedoch wiederum zu einer unnötigen Datenmengenvergrößerung und damit langsameren Bearbeitung führte.

Einen wichtigen Punkt muß man bei dieser Art der Aufnahme jedoch noch beachten: wenn es sich um Halbbilder handelt, die im Interlaced Modus aufgenommen wurden, passen die einzelnen Zeilen von EVEN- und ODD-Feld nicht exakt zusammen. Schließlich bestand das EVEN-Feld ursprünglich aus den Zeilen 0,2,4,6,8... und das ODD-Feld aus den Zeilen 1,3,5,7,9... des kompletten Interlaced-Vollbildes. Wir machen also einen Fehler von vertikal einer halben Zeile, wenn wir die erste Zeile des EVEN-Bildes (ursprünglich die Vollbildzeile 0) mit der ersten Zeile des ODD-Feldes (ursprünglich die Vollbildzeile 1) vergleichen. Dieses Problem ist sehr leicht zu lösen, indem man die Bilder im Noninterlaced-Modus aufnimmt. Die meisten hochwertigen Interlaced-Kameras sind in der Lage, im Noninterlaced-Modus zu arbeiten, wenn man sie entsprechend ansteuert. Die entsprechenden Signale können von der Frame-Grabber-Karte in der Regel erzeugt werden. Es wird dann nur noch jede zweite Zeile der Kamera ausgelesen, allerdings mit doppelter Frequenz, z. B. 50 Hz. Man erhält dann tatsächlich 50 Bilder/Sekunde mit der Auflösung von z. B. 768\*288 Bildpunkten. Dieses Verfahren wird von der Software unterstützt und kann standardmäßig für die Aufnahme schnellerer Bewegungssequenzen benutzt werden.

Bei der Aufnahme vom Videorekorder kann man natürlich nicht auf den Noninterlaced-Modus umstellen: hier ist das Ausgangssignal durch die übliche Videonorm festgelegt und man muß den vertikalen Versatz von einem halben Bildpunkt von EVEN- zu ODD-Feld in Kauf nehmen.

### 5.3. Sequenzaufnahme als Video-Stream direkt auf Disk

Da der Bildspeicher im PC in der Regel begrenzt ist, z. B. 1024 MB oder bei größeren Workstations auch bis zu 8192 MB, kommt der Wunsch auf, eine Videosequenz direkt auf die wesentlich größere Festplatte zu speichern. Hier wären heute 1000GB oder mehr kein Problem. Die Geschwindigkeit heutiger moderner Festplatten ist in der Regel auch hoch genug, zumindest die „normale“ CCIR-Videofrequenz mit 768\*576 Pixeln großen Bildern bei 25Hz mit genügender Sicherheit zu speichern. Dies sind ja nur etwa 11MB/Sekunde. Die Grenze der kontinuierlichen Schreibgeschwindigkeit moderner Festplatten liegt etwa bei 40..80MB/Sekunde. Allerdings kann dies kaum garantiert werden, da erfahrungsgemäß öfters – z. B. alle paar Sekunden – kleine Pausen eingelegt werden, in denen sich etwa die Festplatte synchronisiert oder Windows irgendwelche mehr oder weniger wichtigen Hintergrundprozesse startet. Dazu kommt noch, daß die Platten im äußeren Bereich viel schneller sind als im inneren Bereich. Wenn relativ hohe Geschwindigkeiten verlangt sind, z. B. bei einer Kamera mit 1280\*1024 Pixeln bei 25Hz schon 33MB/Sekunde oder bei der Pulnix TM6710 mit 648\*484 Pixeln bei 120Hz schon 38MB/Sekunde, dann sollte dies bei einem optimierten System jedoch gut funktionieren. Wenn jedoch wesentlich höhere Schreibgeschwindigkeiten verlangt werden, z. B. für richtige Hochgeschwindigkeitskameras, dann sind schnelle RAID0-Plattensysteme oder SSD-Systeme gefragt. Im einfachsten Fall reicht hier schon ein einfaches SATA-Festplatten-RAID-System mit 2 Platten aus, mit dem im Test auf einem einfachen modernen Rechner schon über 100MB/Sekunde erreicht wurden. Mit schnellerem Rechner und 8 Platten kann man bis an die Grenze von etwa 300MB/Sekunde herankommen. Möglicherweise ist hier mit den allerdings auch wesentlich teureren SCSI-RAID-Systemen noch eine gewisse Steigerung zu erreichen. Aber trotzdem scheint bei etwa 350..400MB/s eine Grenze zu liegen, die nur mit speziellen Tricks überschritten werden kann. Darüber hinaus wird es vor allem kostspieliger. Einige auf dem Markt erhältliche Systeme sollen kurz angesprochen werden: mit Systemen der Firma IO-Industries kann eine Schreibgeschwindigkeit von bis zu Gigabyte/Sekunde erreicht werden. Dies sind allerdings externe Systeme, die auch einen speziellen Frame Grabber beinhalten. Weiterhin gibt es neuerdings die SSD-PCIe-Karten der Firma Fusion-IO, mit denen man bis 1.5GB/s beschreiben kann, und das bis zu einer Größe von 2 TB.

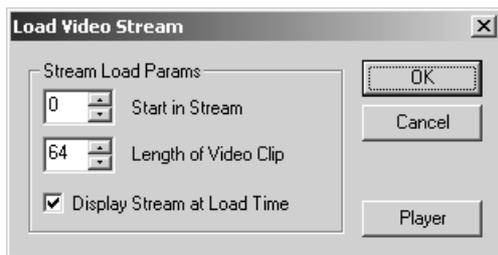
Um die Bilddaten nun kontinuierlich auf die Platte oder das Plattensystem schreiben zu können, wurde im picCOLOR Programm eine Funktion implementiert, die über einen Ring-Speicher eine kontinuierliche Sequenz aufnimmt und diese direkt so schnell wie möglich auf das Plattensystem schreibt. Wenn die Schreibgeschwindigkeit nicht ausreicht, mit der Bildeinlesegeschwindigkeit mitzuhaltend, wird der Schreibzeiger im Ring-Speicher zurückfallen. Dies wird in der Statuszeile als „Delay“ angezeigt. Wenn dies nur ab- und zu einmal vorkommt, wird der Schreibzeiger den Bildeinlesezeiger wieder einholen und man kann auf diese Weise mit einer Videosequenz die gesamte Festplatte vollschreiben. Nur wenn der Schreibzeiger derart weit zurückgefallen ist, daß das gerade wegzuschreibende Bild droht, mit dem nächsten eingelesenen Bild überschrieben zu werden, muß entweder abgebrochen werden oder der Schreibzeiger muß Bilder überspringen. Dies wird in der Statuszeile als „Skip“-Anzahl angezeigt. In diesem Fall sollte die Sequenz allerdings besser abgebrochen werden. Auch wenn die Schreibgeschwindigkeit langsamer ist als die Bildeinlesegeschwindigkeit, kann diese Art der Abspeicherung über den Ring-Speicher sehr wohl sinnvoll sein. Immerhin kann man damit die Länge einer aufgenommenen Sequenz erheblich vergrößern, wenn die Schreibgeschwindigkeit zumindest größer als 50% der Bildeinlesegeschwindigkeit beträgt.

Nach Definition eines Ring-Speichers (hierzu wird einfach eine ganz normale Bildsequenz definiert) und Start der Funktion muß entschieden werden, ob man die Sequenz als kontinuierlichen Video-Stream in einem einzigen File abspeichern möchte oder ob man Einzelbilder als einzelne Files abspeichern möchte. Diese letztere Möglichkeit ist natürlich langsamer, da für jedes Bild ein File geöffnet werden muß, und ist lediglich für

Testzwecke implementiert. Video-Stream-Files dagegen können natürlich sehr lang werden. Da manche Systeme File-Größen von mehr als 2GB nicht zulassen oder bei größeren Files deutlich an Geschwindigkeit verlieren, werden die Stream-Files in picCOLOR auf 1GB begrenzt. Danach wird automatisch ein neuer File mit dem gleichen Namen geöffnet, der sich nur durch eine andere Extension unterscheidet (name.VID, name.V00, name.V01, name.V02, ...). Lediglich an den Anfang des ersten Files wird durch picCOLOR ein File-Header geschrieben, in dem die wichtigste Sequenzinformation enthalten ist. Dieser Header kann mit einem ASCII-Anzeigeprogramm betrachtet werden. Er ist folgendermaßen aufgebaut (hier für eine Bildgröße von 648\*484 Pixeln bei 8Bit Grauwerten):

```
picCOLOR_Video_Stream W=0648 H=0484 B=08 STEREO=0 .....
```

Dieser Header ist genau 64 Bytes lang, danach beginnen direkt die Bilddaten. Wenn das Stereo-Flag in picCOLOR gesetzt ist, werden Stereosequenzen abgespeichert, wobei immer abwechselnd das linke und das rechte Bild gespeichert wird. Die Bildgröße richtet sich streng nach der x/y-Größe des Sequenzspeichers.



Solche abgespeicherte Bildsequenzen können über die „Load Stream“-Funktion wieder eingelesen werden. Da der Stream ja in der Regel länger als ein intern definierter Bildsequenzspeicher ist, kann hierbei ausgewählt werden, wieviele Bilder des Streams und ab welcher Bildnummer des Streams eingelesen werden soll. Während des Einlesens kann die komplette Sequenz angezeigt werden. Hierbei wird die laufende Bildnummer in der Statuszeile angezeigt, um auf diese Weise zumindest in etwa bestimmte Sequenzbereiche finden zu können. Man kann allerdings auch einen einfachen Video-Stream-Player starten, mit dem man die komplette Sequenz langsam und schnell vorwärts und rückwärts betrachten kann und sogar Einzelbilder oder Unter-Sequenzen abspeichern kann.

Durch die Namensgebung mit .VID, .V00, .V01, ..., .V99 ist die Gesamtlänge des Streams auf 101 GB begrenzt. Auf Wunsch kann dies ohne Probleme auf 202 GB vergrößert werden, indem 2GB-Files zugelassen werden. Wenn noch längere Streams gespeichert werden sollen, sprechen Sie bitte das Entwicklungsteam an – durch eine andere Namensgebung ist dies natürlich machbar.

#### 5.4. Hochgeschwindigkeitskameras

Für die Auflösung extrem schneller Bewegungen sind natürlich auch 50 oder 60 Bilder/Sekunde beim CCIR oder RS-170 Signalstandard noch recht knapp. Technische Schwingungsvorgänge haben oft höhere Eigenfrequenzen, so daß zur Auswertung unbedingt wesentlich höhere Aufnahmegeschwindigkeiten benutzt werden müssen. Hier bieten sich einige Spezialkameras an, die über eine spezielle Ansteuerung des CCD-Sensors z. B. bei verringerter horizontaler oder vertikaler Auflösung recht hohe Frequenzen erreichen. Beispiele dafür sind die analoge JAI CV-M30 mit maximal 360 Hz Aufnahmefrequenz bei 768\*67 Pixeln Auflösung, die digitale Pulnix TM6710 mit maximal 350 Hz bei 648\*100 Pixeln Auflösung, oder der Hitachi KP-F102 mit bis zu 190 Hz bei 1300\*??? Pixeln. Damit können schnelle Bewegungen auf einfache und relativ preiswerte Art ausgewertet werden. Bei noch höheren Anforderungen an die Aufnahmegeschwindigkeit muß man schließlich auf eines der echten Hochgeschwindigkeitssysteme übergehen. Hier gibt es entweder Systeme mit eigenem Bildspeicher (z. B. Vosskühler mit 1024\*1024 Pixeln bei 500 Hz, Weinberger mit 1536\*1024 Pixeln bei 1000 Hz oder KODAK, Redlake, Photron bis etwa 5000 Hz bei jeweiligem Vollbildformat) oder die CMOS-Kameras mit Chips der Firma PhotoBit (jetzt von Micron vertrieben) oder der Firma Cypress (mit ihren Lupa-Sensoren), die Busbandbreiten von 660 MByte/Sekunde und mehr erfordern. Da interne Speicher große Nachteile haben, wenn es um Echtzeitauswertungen geht oder wenn lange Bildsequenzen in den PC zur Auswertung übertragen werden müssen, wird man in vielen Fällen die direkt angeschlossene Kamera ohne internen Speicher vorziehen. Um Systeme mit solchen Kameras sinnvoll laufenzulassen, sollten sehr schnelle Frame-Grabber auf Basis des PCI-X-Busses oder noch besser des PCI-Express-Busses eingesetzt werden. Auch auf Basis des etwas abgespeckten PCI-X-Busses mit 64Bit/66MHz können noch beachtliche Geschwindigkeiten erzielt werden. Verschiedene Workstation- oder Server-Mainboards stellen den schnellen PCI-X-Bus zur Verfügung. Der modernere PCI-Express-Bus wird inzwischen von allen Mainboards für Grafikkarten geboten. Durch die Möglichkeit, mehrere Grafikkarten per SLI oder CrossFire Methode einzusetzen, kommen jetzt Gaming-Mainboards auf den Markt, die sogar 4 schnelle PCI-Express Slots aufweisen. Dort könnte man z. B. eine schnelle Grafikkarte, zwei schnelle Frame Grabber und noch einen schnellen RAID-Controller einsetzen. Die

bisher tatsächlich erzielten Transferrgeschwindigkeiten sind wie folgt: Der PCI-X mit 66MHz/64Bit erzielt je nach eingesetztem Frame Grabber bis zu 430MB/s. Der vollständige PCI-X-Bus sollte bis zu 700MB/s erreichen, leider gibt es hierzu jedoch keine Frame Grabber. Mit dem PCI-Express-Bus in der 4-Lane-Version konnten bisher bis zu 700MB/s erreicht werden. Inzwischen sind auch Frame Grabber mit 8-Lane-Busanschluß auf den Markt gekommen, so daß dann 1500MB/s möglich sein sollten. In vielen Fällen wird eine Konfiguration mit zwei Kameras im Stereo-Mode eingesetzt werden. Dann muß beachtet werden, daß die verfügbare Bandbreite nicht geteilt wird. Beim PCI-X-Bus muß man also darauf achten, daß zwei komplett getrennte Busse vorhanden sind. Beim PCI-Express-Bus gibt es in der Regel kein Problem, da der PCI-Express eigentlich kein Bus ist, sondern eher sternförmig an die Northbridge des MainBoard-ChipSets angeschlossen ist – wenn genügend Steckplätze vorhanden sind, sollten sie auch alle die volle Transferrgeschwindigkeit erreichen können.

### 5.5. Farbbildsequenzen und Stereo-Sequenzen

Während bei einem Grauwert-System normale Grauwert-Kameras in der Regel immer angeschlossen werden können, lassen sich 24-Bit Echtfarbbildsequenzen oder Stereosequenzen nur mit geeigneten Frame Grabbern aufnehmen. Wenn der eingesetzte Frame Grabber die hierfür notwendigen Funktionen zur Verfügung stellt, so werden die Farb- oder Stereoanteile gleichzeitig aufgenommen und automatisch in die richtigen Speicherbereiche geleitet. Wenn ein einfacher Frame Grabber jedoch nur gemultiplexte, also umschaltbare Videoeingänge besitzt, so gibt es trotzdem die Möglichkeit, Farb- oder Stereobilder aufzunehmen: die gemultiplexten Eingänge werden so schnell wie möglich nacheinander umgeschaltet und die Teilbilder eingelesen. Hierbei erreicht man bei 24-Bit Farbbildsequenzen mit CCIR-Signal z. B. 8.3333Hz, also 1/3 Echtzeit, bei Stereosequenzen mit zwei Kameras 12.5Hz, also halbe Echtzeit.

Für Stereo-Sequenzen müssen zuvor zwei Sequenz-Bildspeicher definiert werden und in der „Camera Selection“-Dialogbox bzw. in der „Stereo“-Dialogbox festgelegt werden. Zusätzlich muß das normale Stereo-Flag gesetzt sein und es muß außerdem der Schalter „Two Screen Acquisition“ eingeschaltet sein. Wenn eine solche Stereo-Sequenz mit dem Video-Player angezeigt wird, so werden auch beide Stereo-Bildspeicher aktiviert, d. h. nach vorne gestellt. Als Benutzer muß man allerdings selbst dafür sorgen, daß sie auch sichtbar sind, d. h. sich nicht gegenseitig überdecken.

Farbsequenzen mit 3 Farbraumanteilen (RGB) werden von picCOLOR z. Zt. nicht unterstützt. Wenn trotzdem Farbbildsequenzen aufgenommen werden sollen, so kann dies mit einem normalen Sequenz-Bildspeicher geschehen, wenn die Kamera die Signale im Bayer-Pattern-Verfahren als kodiertes Grauwertbild liefert. Solche Bilder können dann nachträglich in RGB-Farbbilder umgerechnet werden – die Funktionen hierzu stellt picCOLOR zur Verfügung.. Auf diese Art ist es auch möglich, Farb-Stereo-Sequenzen einzulesen.

### 5.6. Einlesen von 3D-Bildern

Ein 3D-Bild wird sicherlich meistens in der Form von Einzelbildern, d. h. Schnitten eingelesen. Wichtiges Beispiel ist hier die Mikroskop-Fokusserie. Hierbei wird ein Objekt unter dem Mikroskop - in der Regel ein mehr oder weniger durchsichtiges Objekt - angefahren und als 2D-Schnitt eingelesen. Dann wird der Z-Antrieb ein wenig verstellt, so daß eine andere Ebene scharf abgebildet wird, und es wird das nächste Schnittbild übernommen usw. Dieses Verfahren kann natürlich auch automatisiert werden, indem man über eine MACRO-Funktion und eine Rechnerschnittstelle den Z-Antrieb des Mikroskopes ansteuert.

Für den Fall, daß ein 3D-Bild innerhalb möglichst kurzer Zeit eingelesen werden soll, z. B. weil sich das Bild langsam verändert, ist das schnelle Einlesen wie bei einer Bildsequenz sinnvoll. Hierzu muß dann allerdings eine Triggerung auf den Verfahrensvorgang, also z. B. den Z-Antrieb des Mikroskopes, gewährleistet sein, so daß nicht gerade verfahren wird, während die Kamera das Bild aufnimmt. Die Sachlage vereinfacht sich, wenn die Kamera auf sehr kurze Verschlusszeiten eingestellt werden kann.

Größere durchsichtige 3D-Räume können z. B. mit einer bewegten Lichtschnittoptik aufgenommen werden. Hierzu muß die Bewegung des Lichtschnittes auf die Aufnahme Frequenz synchronisiert werden. Solche Aufnahmetechniken wären sinnvoll für die PIV-Auswertetechnik (Particle Image Velocimetry) zur Vermessung von Strömungsfeldern in Wasser oder Luft mit winzigen schwebenden Partikeln. Siehe hierzu auch das PIV-Zusatzmodul.

### 5.7. Einlesen mit arithmetischer/logischer Funktion

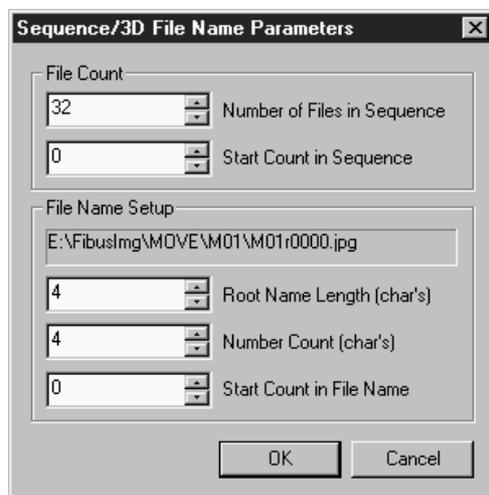
Bei hardwaremäßiger Eignung der Bildeinlesekarte kann auf Wunsch programmiert werden, daß die Übertragung vom Bildschirmspeicher in den Arbeitsspeicher mit einer logischen oder arithmetischen Funktion zwischen Live-Bild und einem bereits gespeichertem Bild durchgeführt wird. Hiermit könnte man z. B. einen störenden Hintergrund schon während der Aufnahme automatisch ausblenden. So bietet die ältere CORECO F64P-Karte eine Echtzeit-Hardware an, solche arithmetische Funktionen direkt mit dem Video-Strom auszuführen. Eine Implementation kann auf Wunsch ausgeführt werden. Bei einigen der DALSA-Karten kann ein spezieller Pixel-Prozessor geliefert werden, mit dem sogar recht komplexe Filter in Echtzeit während der Bildübernahme ausgeführt werden können. Mit der Micro-Enable-Karte ist dies über eine spezielle Programmierung des FPGA-Chips ebenfalls möglich.

## 6. Ein- und Ausgabe auf Festplatte

3D-Bilder und Sequenzen sind im Arbeitsspeicher des picCOLOR-Systemes immer als Einzel- oder Schnittbilder gespeichert. Dies vereinfacht den Zugriff und ermöglicht es, mit den normalen Lade- und Speicherroutinen und Dateiformaten zu arbeiten. In der Regel wird jedes Einzelbild in einem gesonderten File gespeichert und erhält einen beliebigen eigenen Namen. Normalerweise sollte dies ein Wurzelname (Root-Name) mit laufender Nummer sein. Die Einzelbildabspeicherung oder das Hereinladen der Bildsequenzen in hunderten oder tausenden von Einzelbildern ist natürlich nicht mehr interaktiv von Hand durchzuführen. Hierzu stehen spezielle Funktionen zur Verfügung, die nach Eingabe des Verzeichnisses, des Wurzel-Namens und einiger weiterer Parameter die Lade- oder Speicherfunktion automatisch übernehmen. Zusätzlich werden hierzu auch noch automatisch arbeitende MACRO-Funktionen beschrieben, mit denen man auch Spezialeffekte beim Laden oder Speichern erzielen kann. Nachteil dieser Methode ist natürlich, daß die Verzeichnisse auf der Festplatte recht schnell voll werden und daß außerdem durch die recht großen Speicherblöcke auf Festplatten von z. B. 4kByte im Mittel ein halber Block, also 2kByte pro Bild verloren gehen. Bei kleinen Bildausschnitten und dem stark komprimierenden JPEG-Format kann es dann vorkommen, daß Einzelbilder in der gleichen Größenordnung liegen und daher dann die gesamte Sequenz auf der Festplatte den doppelten Speicherplatz belegt. Man sollte es daher zur Regel machen, längere Sequenzen oder kleine 3D-Bilder mit großer Tiefe in z-Richtung bei längerem Nichtgebrauch mit einem Archivierungsprogramm zu einem einzigen File zusammenzufassen. Gut bewährt haben sich hier die Archivierungsprogramme LHA.EXE oder das bekanntere Programm PKZIP.EXE unter DOS, oder die Programme WinZip oder WinRar unter Windows.

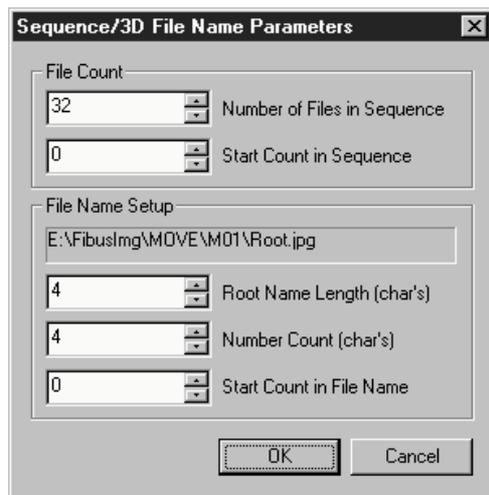
Als Speicherformat für die Einzelbilder steht unter anderem das weit verbreitete TIF-Format zur Verfügung sowie das bei den großen Datenmengen sehr sinnvolle, da stark komprimierende JPEG-Format. Eine Kompression auf 10 bis 20% der ursprünglichen Datenmenge ist bei kaum sichtbarer, jedoch meßbarer (!) Bilddegradierung möglich. Wenn eine Bilddegradierung nicht zulässig ist, weil man z. B. genaue Messungen in den Bildern vornehmen will, so sollte man im TIF-Format abspeichern und kann dann bei der Archivierung mit den üblichen Archivierungsprogrammen mit der Huffman-Kodierung immerhin doch noch bis zu 50% Platz sparen. Alle weiteren vom picCOLOR-Programm zur Verfügung gestellten Bildspeicherformate (BMP, EPS, PPM) können natürlich auch benutzt werden.

### 6.1. Lade 3D/Sequenz-Bild von Festplatte



Zum Laden eines 3D- oder Sequenzbildes, das den picCOLOR-Namenskonventionen genügt, welches also aus einem Wurzelnamen und einer laufenden Nummer mit gleicher Stellenanzahl besteht (also z. B. Test0001, Test0002, ..., Test0999, ...), kann die vorgegebene Ladefunktion benutzt werden. Zum Laden eines 3D-Bildes oder einer Sequenz muß zuvor der 3D/Sequenz-Speicher richtig eingestellt und definiert werden, d. h. mit der richtigen x,y-Ausdehnung und der richtigen Länge. Größer schadet nicht, bei kleinerem Speicher wird das eingelesene Bild abgeschnitten. Dann wird die „Load 3D/Sequenz“-Funktion im „3D/Sequenz“-Menü oder aus der „Tool-Box“ aufgerufen. In der üblichen Windows-„Open-File“-Dialogbox wird irgendein Bild der Sequenz ausgewählt und geöffnet. Hierbei ist es völlig egal, welches Bild der Sequenz genommen wird. Es erscheint dann eine weitere Dialogbox, in der bestimmte Parameter zur Namenskonvention eingegeben werden können und in der zusätzlich bestimmt werden kann, an welchem Schnittbild des 3D/Sequenzbildes das Laden starten soll. Wichtigste Eingabe ist hier die Länge von Wurzelnamenanteil und von Nummernanteil im Filenamen, die z. Zt. leider nur manuell eingestellt werden können, nicht automatisch.

## 6.2. Speichere Sequenz/3D-Bild



Wie beim Laden eines 3D/Sequenzbildes wird zuerst ein Wurzelname in der üblichen Windows-„Save-File“-Dialogbox definiert, dann wird in einer weiteren Dialogbox die genaue Namenskonvention definiert, d. h. Länge des Wurzelnamenanteils und des Nummernanteils. Hierbei überschreiben die angegebenen Zeichenanzahlen die Angabe des Wurzelnamens. Das bedeutet folgendes: wenn als Wurzelname Test0001 eingegeben wurde, so wäre es z. B. sinnvoll als Wurzelnamenlänge 4 und als Nummernlänge ebenfalls 4 einzugeben. Wenn man jedoch für die Wurzelnamenlänge eine 2 eingibt, werden die Filenamen nur mit den ersten zwei Zeichen des Wurzelnamens gebildet, also Te0001, Te0002, usw. Außerdem kann ein Bereich der Sequenz oder des 3D-Bildes bestimmt werden, der abgespeichert werden soll. Die Nummern im Filenamen müssen nicht notwendigerweise den Nummern des Schnittbildes im Sequenzspeicher entsprechen.

## 6.3. Laden und Abspeichern von Stereo-Bildsequenzen

Zum Laden und Speichern von Stereo-Bildsequenzen können die normalen Sequenzfunktionen verwendet werden. Dies ist allerdings umständlich und fehlerträchtig, da ja nacheinander die Bildsequenzen für die linke und dann für die rechte Kamera aufgerufen werden müssen. Hierzu muß zuvor der jeweilige Sequenz-Bildspeicher aktiviert werden. Die Gefahr ist natürlich groß, daß man versehentlich den falschen Bildspeicher aktiviert und dann die gleiche Sequenz für links und rechts zweimal abspeichert.

Hier zeigte es sich, daß die Verarbeitung über ein Macro-Programm einfacher und sicherer ist. Mit solch einem Macro-Programm kann dann übrigens auch gleichzeitig der Zeitstempel in einen Datenfile gespeichert werden oder von einem solchen gelesen werden, der bei der Bildaufnahme jedem Schnittbild der Sequenz aufgeprägt wird. Die jeweiligen Macro-Programme sind: LoadStereoSeqTime.MCR und SaveStereoSeqTime.MCR. Sie werden auf der Installations-CD mitgeliefert und können natürlich auch an eigene spezielle Anforderungen angepaßt werden.

## 6.4. Abspeichern von Sequenzen im AVI-Format

Im Unterschied zu üblichen Multi-Media-Systemen erfolgt beim picCOLOR-Programm die Abspeicherung im Arbeitsspeicher unkomprimiert, damit die Bilder nach der Aufnahme zur direkten und schnellen Verarbeitung zur Verfügung stehen. Beim Speichern auf die Festplatte können die Bilder natürlich komprimiert werden - mit dem JPEG-Verfahren leicht bis auf 10 oder 20% der ursprünglichen Größe. Wenn jedoch die Bilder später quantitativ ausgewertet werden sollen, so sollte man vor der Komprimierung genau prüfen, ob nicht wichtige Meßwerte, z. B. Objektkanten, durch die Kompression verändert werden und so nach dem Dekomprimieren zu Meßfehlern führen. Besonders bei Subpixelauswertungen ist eine Kompression auf keinen Fall anzuraten.

Eine noch weitergehende Kompression bis zu einem Faktor von 100 kann durch Abspeicherung als MPEG-Video erreicht werden. Dann läßt jedoch die Qualität der Bilder deutlich nach und eine meßtechnische Auswertung ist kaum noch sinnvoll.

PicCOLOR erlaubt die Abspeicherung im AVI-Format, allerdings nur unkomprimiert. Nachträglich kann man jedoch leicht einen AVI-File komprimieren, indem man frei verfügbare Programme wie z. B. VirtualDub benutzt. VirtualDub ist in der Lage, alle in Windows implementierten Filter bei der Kompression zu benutzen. Beim Speichern des unkomprimierten AVI-Files muß jedoch folgendes beachtet werden: AVI-Files werden je nach Verfahren in verschiedenen Farbräumen kodiert, z. B. RGB oder YUV422. PicCOLOR benutzt das einfache unkomprimierte RGB-Format. Grauwertsequenzen werden dabei in jedem Fall um den Faktor 3 aufgebläht. Die implementierten Routinen sind noch nicht optimiert, es wurden intern teilweise noch 16Bit Integer-Variable verwendet, die bei zu langen Files einen Overflow erzeugen können. Daher muß z. Zt. die AVI-

File-Größe auf 1GB begrenzt werden. Wenn Sequenzen länger werden, wird eine Fehlermeldung ausgegeben. Man kann dann die Sequenz in kürzeren Bruchstücken abspeichern und später wieder zusammenfügen. Hierzu müssen einfach die „Start“ und „End“ Regler im Rekorder oder Video-Player eingestellt werden.

Man kann auch aus gespeicherten Einzelbildern AVI oder MPEG-Streams erzeugen. MPEG-Kodierprogramme können auf dem Internet gefunden werden. Recht ordentlich und außerdem frei zugänglich ist z. B. das Programm "mpeg\_enc.exe" (oder komplett mit Beschreibung und Beispiel "ENC11DOS.ZIP") der MPEG Society (Info: MPEG-L@netcom.com oder Download: ftp.netcom.com:/pub/cfogg/mpeg2), das auch von Phade Software (phade@tu-berlin.de oder phade@contrib.de) gepflegt wird. Leider ist dieses Programm jedoch ein DOS-Programm. Das "mpeg\_enc.exe"-Programm verlangt als Eingabe Bilder im ".PPM"-Format. Dies ist ein in der Unix-Gemeinde häufig verwendetes sehr einfaches Bildformat, das auch vom picCOLOR-Programm zur Verfügung gestellt wird. In der implementierten Form ist es für Farbbilder gedacht, Schwarz/Weiß-Bilder werden bei gleichen RGB-Farbwerten auf das 3-fache aufgebläht. Es ist daher sinnvoll, die PPM-Bilder nach der MPEG-Umwandlung zu löschen.

Es ist geplant, in das picCOLOR-Programm auch Filter für die unter Windows implementierten Codecs zu implementieren. Dann wird es möglich sein, direkt vom picCOLOR-Programm aus hochkomprimierte AVI- oder MPEG-Sequenzen abzuspeichern.

## 7. Verarbeitung von 3D -Bildern

Während die Definition und Speicherung von Sequenzen und 3D-Bildern im picCOLOR-Programm gleich ist und auch die Darstellung in Einzelbildern der Sequenz oder Schnittbildern des 3D-Bildes immerhin sehr ähnlich, ist die Verarbeitung von 3D-Bildern meist grundsätzlich unterschiedlich. In Sequenzen liegen in der Regel zwei Ortsdimensionen vor und eine Zeitdimension, bei 3D-Bildern liegen drei gleichwertige Ortsdimensionen vor. Die Pixel werden hier Voxel genannt, entstanden aus "Volume Pixel Element".

In Sequenzen werden in der Regel Zuordnungen in aufeinanderfolgenden Zeitreihenbildern vorgenommen, z. B. zur Detektion einer Bewegung. In 3D-Bildern dagegen werden z. B. 3D-Filter angewandt oder Objekte im 3D-Raum gesucht. Dies bedeutet jedoch nicht, daß nicht doch einige der Funktionen für beide Anwendungen gleichermaßen sinnvoll eingesetzt werden können.

Da 3D/Sequenz-Bilder in der Regel viel Speicherplatz benötigen, wird eine UNDO-Funktion z. T. nicht zur Verfügung gestellt. Wenn man jedoch eine UNDO-Funktion benötigt und den hierfür notwendigen Speicherplatz etwa bei kleinem 3D-Bild bereitstellen kann, so sollte man ein zweites 3D-Bild definieren und vor jedem kritischen Verarbeitungsschritt das Bild in den zweiten 3D-Speicher durch einen Kopiervorgang retten. Natürlich ist es auch möglich, eine direkte Undo-Funktion einzubauen. Bei tatsächlichem Bedarf wenden Sie sich bitte an das Entwicklungsteam.

Wenn ein normaler 2D-Bildspeicher und damit das normale 2D-Bildspeicher-Hauptmenü aktiviert ist, so erhält man im Processing-Menü unter dem Eintrag 3D-Processing das folgende Menü, mit dem in erster Linie 3D-Schnittbilder in den 2D-Bildspeicher zur interaktiven Bearbeitung transferiert werden können. Hiermit können 3D-Bildschnitte interaktiv oder per MACRO-Programm im 2D-Bildspeicher verarbeitet werden und dann wieder in den 3D/Sequenzspeicher zurückkopiert werden. Per MACRO-Programm ist auch eine direkte zeitsparendere Verarbeitung im 3D/Sequenz-Speicher ohne den Transfer zum 2D-Bildspeicher und zurück möglich. Dabei gibt es jedoch gewisse Einschränkungen wegen der im 2D-Bild eingestellten ROIs (Region of Interest).

Processing (Regular 2D-Buffer) -> Sequence Functions/ 3D-Functions:

Set 3D-Active
Set 3D-Second
Get 3D-Image
Copy 3D(xy)->ROI
Copy 3D(xz)->ROI
Copy 3D(yz)->ROI
Copy ROI->3D(xy)

Hier findet man Funktionen zur Auswahl einzelner 3D/Sequenz-Schnittbilder und Funktionen zur Wahl des aktiven und bei dualen Funktionen (z. B. Kopieren eines 3D/Sequenzbildes auf ein zweites) auch des zweiten benutzten 3D-Bildspeichers.

Wenn jedoch ein 3D/Sequenz-Speicher und damit das spezielle 3D/Sequenz-Hauptmenü aktiviert ist, so erhält man unter dem Hauptmenüpunkt „Processing“ eine Menüstruktur, die der eines 2D-Bildspeichers sehr ähnlich ist, jedoch ausschließlich 3D- und Sequenzbearbeitungsfunktionen zur Verfügung stellt:

Processing (3D/Sequence-Buffer):

Cut	
Copy	
Paste	
-----	
Select Active 3D/Seq.	
Select Second 3D/Seq.	
-----	
Copy 3D/Seq.	
3D-Geometry	->
3D-Arithmetic	->
3D-Filters	->
3D-Frequency	->
3D-Binary	->
3D-Morphology	->
3D-Miscellaneous	->

Dies sind hauptsächlich Funktionen zur Vorverarbeitung und Segmentierung von Sequenzen oder 3D-Bildern. Prinzipiell können jedoch beinahe alle Funktionen der 2-dimensionalen picCOLOR-Bibliothek auch als 3D-Funktionen geschrieben werden. Bei speziellen Wünschen bitte mit dem Entwicklungsteam Rücksprache halten.

### 7.1. Kopiere 3D-Schnittbild -> Bildspeicher

Die Funktionen zum Kopieren eines Schnittbildes aus dem 3D/Sequenz-Speicher und zurück sind nur vom normalen 2D-Bild-Menü aus erreichbar (siehe Kap. 4.1. und 4.2. des Haupthandbuches). Während bei Sequenzen üblicherweise ein Bild nach dem anderen in den Bildspeicher übertragen wird und dort mit 2-dimensionalen Funktionen verarbeitet wird und dann eventuell wieder in den Sequenz-Speicher zurückkopiert wird, werden die meisten 3D-Verarbeitungsfunktionen direkt im 3D-Speicher ausgeführt. Für bestimmte Zwecke kann jedoch die Arbeit im 2-dimensionalen Bildspeicher trotzdem sinnvoll sein. Dazu werden die Funktionen "Copy 3D->ROI" und "Copy ROI->3D" zur Verfügung gestellt. Die Kopierfunktionen vom 3D-Speicher in die aktuelle ROI existieren in drei verschiedenen Versionen: es können xy-Ebenen, yz-Ebenen und zx-Ebenen kopiert werden.

### 7.2. 3D-Geometry

Zur Bearbeitung der 3D-Geometry sind noch keine direkten Funktionen vorhanden. Man kann selbstverständlich die vorhandenen 2D-Funktionen verwenden, z. B. für horizontales oder vertikales Spiegeln, für Rotationen, usw.

### 7.3. 3D-Arithmetik

Folgende einfache arithmetische Funktionen für ein 3D-Bild oder für zwei 3D-Bilder sind implementiert :

- Löschen des 3D-Bildes (über „Clear 3D/Sequence“-Funktion im 3D/Sequenz-Menü)
- Invertieren (Jedes Voxel (von Volume Pixel Element) wird zu [255-Grauwert] gesetzt)
- Addieren oder Subtrahieren, Multiplizieren oder Dividieren einer Konstanten
- Scaling, Quantify, Power mit Konstanter
- Logische Funktionen mit Konstanter (AND, OR, XOR, NAND, Shift)
- Kopieren eines 3D-Bildes auf ein anderes
- Logische Funktionen zwischen zwei 3D-Bildern (AND, OR, XOR, usw.)
- Arithmetische Funktionen mit zwei 3D-Bildern (MAX, MIN, ADD, SUB, MUL, DIV, usw.)
- Energiebestimmung von zwei Bildern (SQRT(Bild1\*Bild1+Bild2\*Bild2))
- Zyklisches Subtrahieren von zwei Bildern (Mit und ohne Benutzung des Null-Wertes)

### 7.4. 3D-Filter

Folgendes Subset der 2D-Filter Funktionen sind für 3D-Bilder implementiert:

- Average 3\*3\*3
- Gauss 3\*3\*3
- Laplace Ultra 3\*3\*3
- Sharpen, d.h. Bildverschärfung mit Addition von Originalbild und Laplace-Kantenbild, zwei Stufen
- Sobel (Sobel-Hx, Sobel-Hy, Sobel-Hz, jedoch noch kein Sobel-Betrag)
- Median-Filter ist in der 3\*3\*3-würfelförmigen und in der 5\*5\*5-kugelförmigen Version implementiert
- Rank-Filter stehen als Funktionen mit 3\*3\*3-Würfel-Kernel und dem kugelförmigen 5\*5\*5-Kernel zur Verfügung. Hierbei ist der 5\*5\*5 Kernel wie ein Würfel, aber die Eck-Voxel werden nicht benutzt.

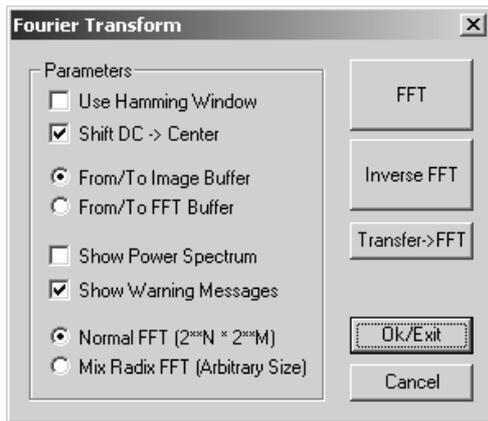
### 7.5. 3D-Morphologie

- Erosion/Dilation

Die Erosions- und Dilationsfilter werden mit den Rank-Filtern gebildet. Diese stehen als Funktionen mit 3\*3\*3-Würfel-Kernel und dem kugelförmigen (Würfel ohne die Eck-Voxel) 5\*5\*5-Kernel zur Verfügung.

### 7.6. 3D-Frequenzraumverarbeitung

Im Unterschied zu den 2D-Bildspeicher-Transformationen muß vor der Ausführung der 3D-FFT ein 3D-FFT-Speicher explizit definiert werden. Hierzu muß die Kantenlänge des zu bearbeitenden Würfels eingegeben werden [16,32,64,128...]. Funktionen beliebiger Kantenlänge sind noch nicht implementiert, was auf Wunsch jedoch nachgeholt werden kann. Das zuvor definierte 3D-Bild muß in allen Dimensionen gleich oder größer als diese Kantenlänge sein. Wenn es größer ist, so wird die FFT im Unter-Bildwürfel ab der vorderen linken oberen Ecke berechnet. Für jeden definierten 3D-Bildspeicher kann ein 3D-FFT-Speicher eingerichtet werden. Da ein 3D-FFT-Speicher wegen der Single-Precision-Floating-Point-Darstellung für komplexe Zahlen den 8-fachen Speicherplatz eines 3D-Bildspeichers benötigt, ist vorgesehen, daß man diesen auch explizit wieder freigeben kann.



#### 7.6.1. Transformationsfunktionen

- 3D-FFT-Transformation
- 3D-IFFT (inverse Transformation)
- Transformation mit oder ohne Hamming Window
- Transformation mit DC-Anteil in der Mitte oder in der vorderen linken oberen Ecke

#### 7.6.2. 3D Spectrum

Berechnung von Amplitudenspektrum, Phasenspektrum, Realteil oder Imaginärteil oder der Umrechnung im FFT-Buffer von der Real/Imaginär- in die Power/Phase-Darstellung und zurück.

#### 7.6.3. 3D Frequency Filters

- Tief-, Hoch- und Bandpassfilter
- Gaussian Bandpass
- Gaussian Smooth
- Gabor Filter
- Difference of Gaussian
- Inverse oder Wiener Filter

#### 7.6.4. 3D Frequency Arithmetics

- Clear FFTBuffer
- Add/Sub Compl.Const
- Mul/Div Compl.Const
- Conjugated Complex
- Inverse Complex
- Normalize Complex
- Add: Act = Act + Scnd
- Add: Act = Act + Scnd.Re
- Add: Act = Act + Scnd.Im
- Sub: Act = Act - Scnd
- Mul: Act = Act \* Scnd
- Mul: Act = Act \* Scnd.Re
- Mul: Act = Act \* Scnd.Im
- Div: Act = Act / Scnd
- AutoCorrelation Active
- X-Correlation Active and Second

#### 7.6.5. 3D-Polynomial-OTF

Erzeugung einer theoretischen 3D-Polynomial-OTF (Optische Transferfunktion). Ein defokussiertes Bild kann als scharfes Bild und dessen Konvolution mit der Punktverbreiterungsfunktion (Point Spread Function: PSF) des defokussierten Linsensystemes angesehen werden. Die Fouriertransformierte der PSF ist die Optische Transferfunktion (OTF). Da die Konvolution im Ortsbereich im Frequenzbereich der Multiplikation entspricht, kann das Bild gefiltert, d. h. korrigiert werden, indem man die Fouriertransformierte des Bildes mit dem Inversen der OTF multipliziert und dann in den Ortsbereich zurückwandelt (Inverse Filterung). Im 3-dimensionalen Raum ist dies besonders interessant, wenn teilweise durchsichtige Objekte unter dem Mikroskop als 3-dimensionale Schnittbilder aufgenommen werden sollen: auch wenn die Tiefenschärfe eines Mikroskopes sehr gering ist, enthält jeder fokussierte Schnitt durch das Objekt die unscharfe Information aller anderen nicht fokussierten Schnitte. Über die Multiplikation mit dem Inversen der OTF kann diese störende Information stark reduziert

werden. Leider ist dieses Verfahren nur dann wirksam, wenn die tatsächliche OTF des Mikroskopes der theoretischen sehr nahe kommt. Es besteht allerdings auch die Möglichkeit, die tatsächliche OTF experimentell zu ermitteln. Hierzu muß jedoch ein bei der gewünschten Vergrößerung derart winziges Objekt aufgenommen werden, daß es nur etwa einen Pixel groß ist.

### 7.7. 3D-Binärisierung

Z. Zt. ist lediglich eine einfache Binärisierung von 3D-Bildern über einzugebene Schwellwerte implementiert.

- Threshold
- Band (lower and upper threshold)

Die meisten weiteren Binärisierungsverfahren aus der 2D-Bildverarbeitung lassen sich jedoch in 3D umsetzen, wenn hierzu tatsächlicher Bedarf besteht.

### 7.8. 3D-Vermessungsfunktionen

Neben den einfachen interaktiven Pixel (oder Voxel)-Vermessungen direkt im 3D/Sequenzbild mit dem Kursor-Fadenkreuz können auch komplexe Vermessungen durchgeführt werden. Solche Funktionen würden im picCOLOR-Measurement-Menü aufgeführt werden. Z. Zt. sind hier jedoch keine speziellen Funktionen implementiert.

Selbstverständlich können 3D-Vermessungen jedoch über MACRO-Programme durchgeführt werden. Ein sehr einfaches Beispiel ist die 3D-Volumenvermessung nach erfolgter Segmentierung und Binärisierung: hierbei wird die Objekt-Fläche (helle Objekte) in den Schnittbildern bestimmt und eine Volumenbestimmung über die vorher einzugebende Schichtdicke durchgeführt.

```

/*****\
* D3D_VOLUME.MCR      Volume Measurement in 3D Image Buffer
\*****/
transfer_to_undo -1      /* set up 2D Undo Buffer, transfer last image */
enable_undo 0           /* no change anymore in Undo buffer */

get_3d_image            /* get information about 3D-buffer */
roi_size IVAR[10] IVAR[11] 0 0 /* set 2D-ROI to 3D-buffer size */
IVAR[1] := IVAR[12]      /* initialize loop variables, here number of slices */
IVAR[2] := 0            /* start with slice 0 in 3D-Buffer */
FVAR[1] := 0.0          /* initialize the sum of 2D-areas in all slices */

float_input "Input depth of slices [float, in pixel thickness]"
FVAR[2] := FVAR[10]     /* save thickness in float variable 2 */

FOR IVAR[1]             /* loop for all 3D-slices */
  d3d_to_roi (0,IVAR[2]) /* transfer slice to 2D-buffer */
  calc_area 0,255,0     /* calculate all white pixel */
  FVAR[1] += FVAR[10]   /* and sum them up in float variable 1 */
  IVAR[2]++           /* increase slice count */
ENDFOR

FVAR[1] *= FVAR[2]     /* multiply with slice thickness */
STRING[2] := "Volume of all objects = "
STRING[2] := CONCAT ( STRING[2] ,FTOA(FVAR[1]),-1)
STRING[2] := CONCAT ( STRING[2] ,“ voxel“, -1)
user_message (STRING[2]) /* and output result */
enable_undo 1         /* back to normal undo behaviour */

```

### 7.9. 3D-MACRO-Programme

Einige weitere Funktionen zur 3D-Bildverarbeitung sind bereits als MACRO-Programme realisiert. Weitere können selbst erstellt werden oder auf Wunsch auch bereitgestellt werden. Hier ein Beispiel:

- Shift-DC über MACRO-Funktion: Verschiebe DC-Anteil eines Fourier-Transformierten Bildes von der linken oberen vorderen Ecke in die Mitte des 3D-Bildes. Bei einigen Funktionen im Frequenzbereich kann es aus mathematischen Gründen vorkommen, daß die einzelnen 3-dimensionalen Fourierkoeffizienten abwechselnd positiv und negativ sind. In diesem Fall verschiebt sich der Nullpunkt des Spektrums (der DC-Anteil) oder - nach der Rückwandlung - des Ortsbereiches um die halbe Fensterbreite, d. h. was vorher in der Mitte des 3D-

Bildes war, befindet sich nach der Umrechnung über Routinen im Frequenzbereich in der vorderen linken oberen Ecke. Durch einfache Ortsverschiebung über ein MACRO-Programm kann dies rückgängig gemacht werden.

```
/******\
* D3D_SHIFT_DC.MCR      Shift DC in 3D Image Buffer
\*****/
transfer_to_undo -1
enable_undo 0
roi_size 64 64 0 0      /* this MACRO is for 128*128*128 FFT buffer */
dft2_if2 1 0 0 0
IVAR[1] := 32
IVAR[2] := 32

FOR 33
  d3d_to_roi (0,IVAR[2]) /* transfer slice to 2D-buffer */
  shift_space /* use regular 2D-function to shift DC */
  IF IVAR[2] < 64 /* and transfer slice back, but to the correct slice */
    roi_to_d3d (IVAR[2])
  ELSE
  ENDIF
  roi_to_d3d (IVAR[1])
  IVAR[2]++
  IVAR[1]--
ENDFOR
enable_undo 1
```

Diese Funktion wurde indes als wichtig genug erachtet, fest ins Programm implementiert zu werden und kann im Menü gefunden werden.

## 8. Verarbeitung von Sequenz-Bildern

Da Sequenzen in der Regel aus 2-dimensionalen Zeitreihenbildern bestehen, werden üblicherweise alle Einzelbilder mit den 2-dimensionalen Funktionen des picCOLOR-Programmes bearbeitet. Hierzu wird das gewünschte Zeitreihenbild in den aktiven Bildspeicher geholt, dort verarbeitet, und bei Bedarf schließlich zurück in den Sequenzspeicher kopiert. Diese Bearbeitungen werden normalerweise über MACRO-Programme automatisch für alle Einzelbilder der Sequenz durchgeführt.

In bestimmten Fällen ist es jedoch notwendig, Funktionen mit allen Bildern der Sequenz gleichzeitig auszuführen oder zumindest in Zwischenrechnungen die Information aller Bilder zur Verfügung zu haben. Ein gutes Beispiel hierfür ist das Mitteln aller Bilder einer Sequenz, das sehr gut zur Hintergrundkorrektur bei Bewegungen eingesetzt werden kann. Dazu kommen noch Funktionen, bei denen durch spezielle Programmierung eine erhebliche Geschwindigkeitssteigerung z. B. mit den BitBlock-Transfer-Befehlen des Prozessors erzielt werden kann. Hierzu gehören z. B. Kopierfunktionen und bestimmte logische und arithmetische Funktionen.

Da Sequenzen sehr viel Speicherplatz benötigen, wird eine UNDO-Funktion z. Zt. nicht zur Verfügung gestellt. Wenn man jedoch eine UNDO-Funktion benötigt und den hierfür notwendigen Speicherplatz bereitstellen kann, z. B. bei kleinen und kurzen Sequenzen, so sollte man eine zweite Sequenz definieren und vor jedem kritischen Verarbeitungsschritt das Bild in den zweiten Sequenz-Speicher durch einen Kopiervorgang retten.

Die im folgenden beschriebenen Funktionen stehen bereits zur direkten Verarbeitung einer Sequenz zur Verfügung. Prinzipiell können jedoch alle Funktionen der 2-dimensionalen picCOLOR-Bibliothek zur Verarbeitung der Sequenzen verwendet werden. Außerdem sind auch einige der 3D-Funktionen (z. B. die Binärisierung, das Invertieren und weitere Arithmetik-Funktionen) sinnvoll für die Sequenzbearbeitung einsetzbar. Bei speziellen Wünschen halten Sie bitte Rücksprache mit dem Entwicklungsteam.

### 8.1. Kopiere Sequenz-Schnittbild auf den 2D-Bildspeicher

Ausgehend von einem normalen 2D-Bild erhält man unter dem Hauptmenüpunkt „Processing“ das folgende „Sequence Processing“-Untermenü:

Set Active Sequence
Set Second Sequence
ROI -> Sequence (x,y)
Pick Seq.-Image
Seq. (x,y) -> ROI
Seq. (x,time) -> ROI
Seq. (y,time) -> ROI

Hiermit können Sequenz-Zeitreihenbilder in die normalen 2D-Bildspeicher übertragen und zurückkopiert werden. Interessant kann hier auch die Darstellung über der Zeitachse sein, die dann eventuell sogar wiederum mit 2D-Funktionen ausgewertet werden kann. Weiterhin findet man hier Funktionen zur Wahl des aktiven Sequenzspeichers (bei dualen Funktionen auch eines zweiten Sequenzspeichers).

### 8.2. Sequence Processing Functions

Bei aktivem 3D/Sequenz-Bildspeicher und damit aktiviertem 3D/Sequence-Menü findet man unter dem Hauptmenüpunkt „Processing“ neben den schon weiter oben angesprochenen 3D-Funktionen einige speziell optimierte Funktionen für die Sequenzverarbeitung.

Copy Active->Second
Rotate Ring Buffer
Average->2D-Image

#### 8.2.1. Copy Active->Destination

Bei zuvor gesetzter aktiven Sequenz und Bestimmungssequenz (Second/Destination) wird der Inhalt der aktiven Sequenz in die Destination-Sequenz kopiert. Dies kann z. B. dazu benutzt werden, eine Undo-Funktion zu emulieren, d. h. eine Sequenz vor einer kritischen langwierigen Verarbeitung für eine spätere Restaurierung zwischenspeichern.

#### 8.2.2. Rotate Ring Buffer

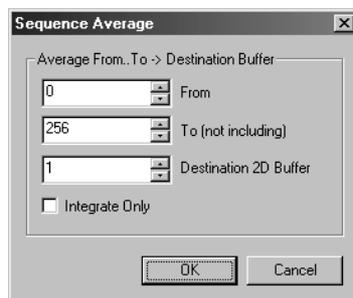
Wie oben bei der Sequenzbildaufnahme beschrieben, kann der Sequenzspeicher als Ringspeicher aufgefaßt werden. Dann wird eine Endlossequenz eingelesen, die mit einer Triggerbedingung (z. B. Maustaste oder ESC) zu beliebigem Zeitpunkt gestoppt werden kann. Auf diese Weise können z. B. einmalig auftretende Vorgänge

gut erfaßt werden. Nach Beendigung der Bildaufnahme liegt die Sequenz dann im Ringspeicher mit irgendeinem Anfangspunkt vor, der dem System natürlich bekannt ist. Wenn die Funktion Rotate-Buffer mit der Defaulteinstellung gestartet wird, d. h. mit dem Parameter "-1", so wird die Sequenz im Sequenzspeicher soweit verschoben, bis ihr Anfang auch am Anfang des Sequenzspeichers beginnt. Das System merkt sich den internen tatsächlichen Sequenzstart solange, bis eine neue Sequenzbildaufnahme gestartet wird, d. h. man kann sich mit dem Verschieben Zeit lassen und z. B. die Sequenz zunächst ansehen, als Videoclip abspielen lassen oder auch einfache Bearbeitungen vornehmen.

In der allgemeinen Version der Funktion, mit positivem Parameter  $1 < n < \text{Sequenzlänge}$ , kann eine beliebige Sequenz um beliebige Einzelbildpositionen verschoben werden, wobei der Sequenzspeicher wiederum als Ringspeicher aufgefaßt wird. Dabei wird rückwärts in Richtung Sequenzanfang geschoben, d. h. bei Parameter "2" wird Bild 2 nach 0 geschoben, Bild 3 nach 1 usw. und Bild 1 zum Ende des Sequenzspeichers.

Die Rotate-Funktion ist z. Zt. noch recht langsam, da hier tatsächlich Speicherinhalte verschoben werden, anstatt lediglich Pointer auf die Schnittbilder zu ändern.

### 8.2.3. Average -> 2D-Image



Mit dieser Funktion kann der pixelbezogene Mittelwert der kompletten Sequenz oder eines Bereiches daraus bestimmt werden und auf den angegebenen normalen 2D-Bildspeicher gespeichert werden.

### 8.3. Sequenzverarbeitung mit MACRO-Programm

Übliche Verarbeitungen von Sequenzen werden über MACRO-Programme durchgeführt. Hierzu werden in der Regel zunächst interaktiv Einzelbilder der Sequenz auf den aktiven normalen 2D-Bildspeicher kopiert und es wird manuell die optimale Verarbeitungsprozedur erarbeitet bzw. entwickelt. Diese kann aus einem oder mehreren komplexen Verarbeitungsschritten bestehen. Es können auch, z. B. zur Bewegungserkennung, jeweils zwei oder mehrere Einzelbilder der Sequenz dazu notwendig sein, die dann in unterschiedliche 2D-Bildspeicher übertragen werden. Nach manuellem Austesten der notwendigen Verarbeitungsschritte kann die gesamte Sequenz dann mit einer MACRO-Funktion automatisch bearbeitet werden. Eine solche MACRO-Funktion kann mit einem beliebigen Text-Editor erstellt werden. Lediglich rudimentäre Programmierkenntnisse sind hierbei notwendig, z. B. die Kenntnis, daß Programme in Einzelschritten oder Einzelbefehlen ablaufen und Schleifen abarbeiten können, in denen etwa der Einzelbild-Dateiname durch Zusammenhängen des Wurzelnamens und der laufenden Nummer zusammengefügt wird. Im Lieferumfang des Sequenz/3D-Modules sind eine Reihe von MACRO-Funktionen enthalten (Files mit der Extension ".MCR"), die wie erwähnt auf einfachste Art vom Benutzer verändert werden können und an spezielle Anforderungen angepaßt werden können. Kopieren Sie einfach die Funktion, die der Gewünschten am nächsten kommt, auf einen File mit neuem Namen und editieren Sie den neuen .MCR-File mit einem beliebigen Texteditor. MACRO-Funktionen werden über das Hauptmenü (User-Menü), mit der entsprechenden Funktion der Tool-Box oder mit der Hot-Key-Taste "M" gestartet.

Als Beispiel wird im folgenden eine solche MACRO-Funktion vorgestellt. Sie prüft zuerst, ob überhaupt ein 3D/Sequenz-Bild definiert ist und paßt dann die aktuelle ROI der normalen 2D-Bildspeicher an die 3D/Sequenz-Größe an. Danach werden nacheinander alle Schnittbilder der Sequenz in die ROI übertragen, dort mit einfachen Funktionen, hier einer Binärisierung und einer anschließenden Umrandungsbestimmung, bearbeitet und anschließend wieder in den 3D/Sequenz-Speicher zurückgespeichert. Prinzipiell ist allerdings auch die schnellere Verarbeitung direkt im Sequenz-Speicher möglich. Dies würde dann die benötigte Zeit zur Übertragung in den 2D-Bildspeicher sparen. Hierbei gibt es jedoch bei bestimmten Funktionen einige Beschränkungen, so daß man diese Möglichkeit mit Vorsicht benutzen sollte. Im folgenden Beispiel wird dann zum Schluß die alte ROI wieder restauriert. Nach Aufruf dieses MACRO-Programmes kann das geänderte Sequenzbild, in dem hier im Beispiel die Konturen der Objekte extrahiert worden sind, mit dem Video-Player angesehen werden. Für eine genaue Beschreibung aller MACRO-Kommandos lesen Sie bitte das MACRO-Handbuch.

```

/*****\
* D3DWORK.MCR Beispiel für eine Sequenz- oder 3D-Bildverarbeitung
\*****/
DEFINT Length          /* Sequenz-Länge: Integer Variable */
DEFINT Index           /* Pointer to Sequence image */

transfer_to_undo -1    /* Setup UNDO-Buffer */
enable_undo 0          /* Do not change UNDO-Buffer any more */
get_roi                /* get ROI coordinates and save them for later usage */
IVAR[40] := IVAR[10]   /* ROI number */
IVAR[41] := IVAR[11]   /* width */
IVAR[42] := IVAR[12]   /* height */
IVAR[43] := IVAR[13]   /* x1 */
IVAR[44] := IVAR[14]   /* y1 */
get_3d_image           /* get 3D/Sequence-image parameters */
IF IVAR[14] # 0        /* if at least one 3D/Sequence-image defined */
  Length := IVAR[12]   /* Length in IVAR[12], save for later usage */
  roi_size IVAR[10] IVAR[11] 0 0 /* adjust ROI for sequence processing */
  Index := 0           /* start with 2D-slice number 0 */
  FOR Length           /* for all 2D-slices of sequence image */
    d3d_to_roi (0,Index) /* copy 2D-xy-slice from sequence to image buffer */

    /* start image processing here */
    binary (3,128,0)    /* processing example: binarization with threshold 128 */
    outline             /* example: finds outlines=object contours */
    /* image processing ends here */

    roi_to_d3d (Index) /* transfer slice back to sequence image */
    Index++           /* point to next 2D-slice */
  ENDFOR
  roi_size IVAR[41] IVAR[42] IVAR[43] IVAR[44] /* restore ROI */
ELSE
  error_box "No 3D/Sequence image defined"
ENDIF
enable_undo 1          /* enable regular UNDO-function */

```

Beachten Sie beim Erstellen von MACRO-Funktionen, daß möglichst nur automatisch ablaufende Funktionen verwendet werden, die keinen Benutzereingriff erfordern. Viele prinzipiell automatisch ablaufende Funktionen können unter bestimmten Bedingungen mehr oder weniger wichtige Meldungen auf den Bildschirm bringen, die dann vom Benutzer akzeptiert werden müssen. Diese Meldungen kann man mit der MACRO-Funktion "msg\_interactive 0" abschalten. Man sollte dabei jedoch bedenken, daß auch wichtige Meldungen auf diese Weise abgeschaltet werden. Im Zweifelsfall sollte man dann vor der Funktion mit eventuell wichtiger Meldung die Ausgabe wieder einschalten (msg\_interactive -1) oder zumindest auf eine bestimmte Zeitkonstante setzen (z. B. msg\_interactive 10 für 10 Sekunden Anzeige und damit auch 10 Sekunden Eingriffs- oder Reaktionszeit). Wenn Sequenzen bearbeitet werden sollen, die auf der Festplatte bereits in Form von Einzelbildern vorliegen, so empfiehlt es sich, dies direkt beim Hereinladen über ein erweitertes Lade-MACRO zu tun. Hierzu muß lediglich in die innere Schleife der Lade-Routine nach dem Laden des Bildes auf den Bildschirm, d. h. in den aktuellen Bildspeicher, die Bearbeitung mit der entsprechenden Funktion des picCOLOR-Programmes erfolgen, bevor das Bild dann in den Sequenzspeicher kopiert wird. Ein kommentiertes Beispiel dazu findet man im folgenden Lade-MACRO-Programm.

```

/*****\
* LOAD3DTF.MCR: Lade 3D/Sequenz-Bild im TIF-Format:
\*****/
DEFINT Count          /* Anzahl der Bilder der Sequenz (Schnitte des 3D-Bildes) */
DEFINT Index1         /* Laufende Nummer des Bildfiles */
DEFINT Index2         /* Index, zeigt auf das aktuelle Bild im Sequenzspeicher */

transfer_to_undo -1   /* Active Image -> Undo-Speicher */
enable_undo 0         /* Undo-Speicher nicht mehr verändern */
get_roi               /* Zwischenspeicherung der aktuellen ROI */
IVAR[40] := IVAR[10]

```

```

IVAR[41] := IVAR[11]
IVAR[42] := IVAR[12]
IVAR[43] := IVAR[13]
IVAR[44] := IVAR[14]

get_3d_image          /* Suche Information über ein definiertes 3D-Bild */
IF IVAR[14] # 0       /* Wenn 3D/Sequenz definiert ist, dann starte... */
  roi_size IVAR[10] IVAR[11] 0 0 /* ROI auf die Größe der definierten Sequenz */

  string_input "Input 3D/sequence path name" /* Einfrage des Pfad-Namens */
  load_pathname STRING[1]
/* statt dessen kann hier vereinfacht auch direkt der Pfad eingegeben werden: */
/* load_pathname "d:\bilder" */

  string_input "Input 3D/sequence file name (Root name only)" /* Rumpfnamen eingeben */
  STRING[2] := STRING[1] /* daran wird später die laufende Nummer angefügt */

  long_input "How many images?" /* Wie viele Bilder hat Sequenz/3D-Bild */
  Count := IVAR[10]

  long_input "Start with image number (for filename)?" /* Startnummer der Filenamen */
/* laufende Nummer für das erste Bild der Sequenz (normalerweise 0) */
  Index1 := IVAR[10]

  long_input "Start with image number (in sequence buffer)?"
/* Ab welchem Bild des Sequenz-Speichers laden ? (normalerweise 0) */
  Index2 := IVAR[10]

/* msg_interactive 0 */
/* Dieser Befehl kann benutzt werden, um unwichtige Fehlermeldungen zu */
/* unterdrücken, die auftreten, wenn TIFF-Bilder aus fremden Programmen */
/* mit unbekanntem und weniger wichtigen TIFF-Befehlen eingelesen werden. */

FOR Count /* Schleife zum Hereinladen aller Bilder der Sequenz */
  STRING[1] := CONCAT (STRING[2], LTOA(Index1), 8)
/* Erstelle Filenamen aus Rumpfnamen und laufender Nummer */
/* mit der '8' wird die Filenamenlänge auf 8 festgelegt. Zwischen */
/* Rumpfnamen und der laufenden Nummer werden Nullen aufgefüllt. */
/* Wenn dies nicht gewünscht ist und die laufende Nummer direkt ohne */
/* Nullen an den Rumpfnamen angehängt werden soll, so wird als */
/* letzter Parameter der CONCAT-Funktion eine '-1' eingegeben. */

  load_filename STRING[1] /* Filenamen dem picCOLOR Programm mitteilen */
  load_tiff /* und Hereinladen des Files */

/* laplace 6 3 */ /* Beispiel für eine Bildverarbeitung beim Laden */
/* der Sequenz: Laplace-Filter */
  roi_to_d3d (Index2) /* und Kopieren des Bildes in den Sequenz-Speicher */
  Index2++ /* Hochzählen der Indizes */
  Index1++
ENDFOR

  roi_size IVAR[41] IVAR[42] IVAR[43] IVAR[44] /* restauriere die alte ROI */

ELSE /* es war kein 3D/Sequenz-Bild definiert, bitte */
/* interaktiv oder MACRO-gesteuert definieren */
  user_message "Define 3D/sequence buffer!"

ENDIF
enable_undo 1 /* Undo wieder enablen, nun kann mit der Taste 'U' der */
/* alte Stand des Arbeitsspeicher wieder restauriert werden. */

```

#### 8.4. Benutzung der 3D-Funktionen für Sequenzverarbeitung

Die arithmetischen Funktionen und die Binärisierungsfunktionen des 3D-Modules arbeiten auf Einzelvoxelbasis - sie können daher auch für die Sequenzbearbeitung genutzt werden. Alle anderen 3D-Funktionen arbeiten mit Voxelnachbarschaften in allen 3 Dimensionen und machen daher normalerweise für eine Sequenz, bei der die Einzelbilder nicht räumlichen, sondern zeitlichen Zusammenhang haben, kaum einen Sinn. Es sind jedoch eine Reihe von Funktionen vorstellbar, die einen räumlichen Bezug ausnutzen könnten: Sie als Anwender sind gefragt, diese Funktionen zu definieren und dem Softwareentwickler mitzuteilen. Ein Beispiel wäre etwa eine Bewegungssequenz mit starken zeitlichen Helligkeitsschwankungen. Solche Helligkeitsschwankungen könnte man durch eine gezielte räumliche Mittelung verringern. Dabei besteht natürlich auch die Gefahr, daß bewegte Objekte weggemittelt werden, so daß es in diesem Fall sicherlich besser ist, eine Angleichung der Mittelwerte der Histogramme aller 2-dimensionalen Schnittbilder zu berechnen. Dies wiederum kann natürlich leicht mit einem MACRO-Programm ausgeführt werden. Bei Interesse wenden Sie sich an das Entwicklungsteam.

#### 8.5. Sequenzvermessungsfunktionen

Sequenzvermessungsfunktionen findet man im picCOLOR-Measurement-Menü. Neben den Kalibrier-Funktionen, die schon aus der 2D-Bildverarbeitung bekannt sind, nämlich der räumlichen Kalibrierung, der Grauwert-Kalibrierung, und der Kamera-Kalibrierung über eine Linsenverzerrung, findet man hier z. Zt. die Funktionen des aufwendigen 3D/Tracking-Modules. Diese sind in einem eigenen Handbuch beschrieben.

## 9. Fehlermeldungen 3D/Sequenz

### 9.1. Fehler bei Definition der 3D/Sequenz-Bilder

"Deallocate 3D/Sequence-buffer(s) first!"	Es wurde versucht, die Sequenzbildgröße zu verändern. Hierzu muß der Sequenzspeicher (oder die Speicher) zuerst freigegeben werden.
"Memory limit: select smaller 3d/sequence buffer"	Der Speicher reicht für die gewünschte Sequenzgröße nicht aus, entweder kleinere x- oder y-Ausdehnung einstellen oder geringere Anzahl einstellen.
"'n' 3d/sequence buffers only!" Sequenz/ Sequenz/	Es kann nur eine Begrenzte Anzahl von (gleichen) 3D-Bildspeichern eingerichtet werden. In den normalen picCOLOR-Versionen sind dies 15 Stück. Dabei kann jede Sequenz maximal 16000 Bilder haben.
"Size of 3d/sequence buffer must not be zero!"	Selbstverständlich darf keine der Dimensionen eines Sequenzbildes oder 3D-Bildes auf Null gesetzt sein.
"Memory limit: select smaller 3D FFT buffer!"	3D-FFT-Speicher benötigen erheblichen Speicher: pro Pixel zwei 4 Byte floating-point-Werte, d. h. 8 Byte
"3D FFT buffer for active 3D image already allocated!"	Für jeden Sequenz/3D-Speicher kann ein einziger FFT-Speicher definiert werden.
"Size of 3D FFT buffer must not be zero!"	Keine Dimension des FFT-Speichers darf gleich Null sein
"Free 3D-FFT Buffers!"	oder
"Dealloc 3D FFT buffer(s) first"	Vor einer Änderung von 3D/Sequenz-Speichergrößen müssen sowohl die 3D/Sequenz-Speicher als auch die 3D-FFT-Speicher entfernt werden

### 9.2. Fehler bei Verarbeitung der 3D/Sequenzbilder

"x-plane not available"	Angewähltes Schnittbild ist nicht definiert.
"y-plane not available"	Angewähltes Schnittbild ist nicht definiert
"z-plane not available"	Angewähltes Schnittbild (3D) oder Einzelbild (Sequenz) ist nicht definiert
"No 3D/Sequence-Buffer Defined"	Es wurde eine 3D/Sequenz-Funktion aufgerufen, ohne daß ein 3D/Sequenzbild definiert ist.
"rank is out of range [1..n] or (-1,0,1)"	Falsche Eingabe bei der Rank-Funktion
"size must be 3, 5, ..., n"	Falsche Eingabe bei verschiedenen Funktionen
"Two 3D-images must be defined"	Bei dualen Funktionen zwei 3D-Speicher definieren
"Active != Second"	Kopieren kann z. B. nicht auf sich selbst erfolgen
"Function will destroy valid 3D-FFT data - ok?"	Gültiger Frequenzspeicher wird überschrieben
"3D-FFT data not valid - transform anyway?" durchgeführt,	Es wurde zuvor keine Fourier-Transformation  das System weiß nicht, ob die Daten im aktuellen FFT-Speicher sinnvoll sind
"No memory for hamming window... Continue anyway? (Left Click)"	Die Hamming-Fensterfunktion benötigt einigen zusätzlichen Speicher. Wenn bei dieser Fehlermeldung trotzdem weitergearbeitet wird, so wird die Filterung über das Hammingfenster bei der FFT-Ausführung einfach weggelassen.
"Function cannot be executed! No 3D-FFT buffer defined!"	Vor Ausführung einer Funktion im Fourier-Speicher muß ein solcher definiert werden
"Requested 3D-FFT buffer not allocated"	Es wurde versucht, einen 3D-FFT-Speicher anzusprechen, der noch nicht definiert ist.
"3D-FFT buffer not defined!"	Vor Ausführung einer Funktion im 3D-Fourier-Raum muß natürlich ein 3D-FFT-Speicher definiert werden.