

## Die AVS-Modulbibliothek RZ-library

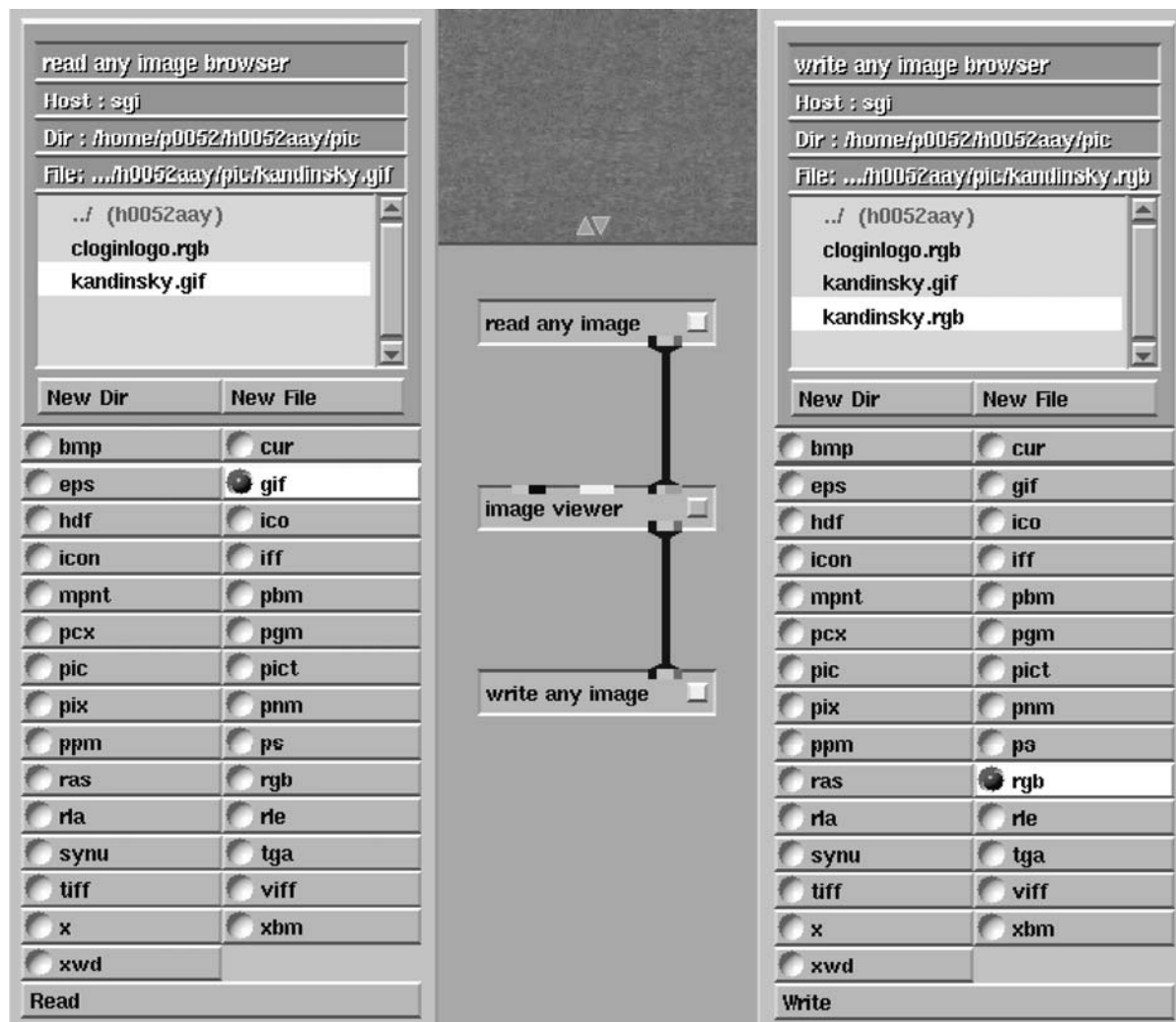
Dieser Artikel beschreibt im groben den Umgang mit der Modulbibliothek *RZ-library* für das Visualisierungssystem AVS. Diese Bibliothek hat noch Prototypcharakter, da sie erst sieben Module enthält. Die von den Modulen angebotenen Funktionalitäten beziehen sich im wesentlichen auf die Arbeit mit verschiedenen Grafikformaten und die Anbindung des digitalen Video-Diskrecorders Abekas A66.

### Laden der Bibliothek

AVS stellt einen Mechanismus zur Verfügung, mit Hilfe dessen man einzelne Module auf einem entfernten Netzwerkrechner ausführen lassen kann. Diese Module werden *remote modules* genannt. Die RZ-library besteht aus *remote modules*, die auf SGI-Workstations arbeiten, da auf diese Weise die Module nur für eine Rechnerarchitektur entwickelt werden müssen. Ruft man AVS auf einer anderen Rechnerarchitektur auf (AVS ist auf allen UNIX-Rechnern des Rechenzentrums verfügbar), lädt man die betreffen-

den Module als *remote modules*. Wir haben uns für die SGI-Workstations entschieden, da wegen ihrer physischen Nähe zum Abekas bei eventuellen Bildübertragungen die Netzwerkbelastung gering gehalten wird.

Um *remote modules* zu laden, müssen AVS zwei Dinge bekannt sein: Wo befindet sich die Modulbibliothek, und auf welchem Rechner soll sie gestartet werden? Diese Dinge lassen sich bei AVS nicht mit einer Kommandozeilenoption angeben. Man muß dazu die Initialisierungsdatei `.avsrc` verwenden. In dieser Datei kann man die Modulbibliotheken angeben, die man laden möchte, sowie einen Pfadnamen für eine `hosts`-Datei. In dieser Datei werden im wesentlichen ein logischer Name für den entfernten Rechner und ein `rsh`-Kommando, mit dem dieser Rechner angesprochen werden soll, festgelegt. Auf die genaue Syntax dieser Dateien wird hier nicht eingegangen. Wir haben unter `/usr/avs/rzlib/etc` den jeweiligen Architekturen angepaßte `avsrc`- und `hosts`-Dateien abgelegt. Diese sollte man sich vor



dem Start von AVS in sein Home-directory kopieren. Danach kann man sie editieren und beliebig mit den einzelnen Optionen experimentieren. Die Einträge für die Modulbibliotheken und die *hosts*-Datei sind in den Beispieldateien gesondert gekennzeichnet.

### Arbeiten mit den Modulen

Nachdem man AVS mit den genannten Einstellungen gestartet hat, sollte man den *Network Editor* laden. Neben den bekannten Modulbibliotheken wie *Volume*, *Chemistry* und *Imaging* müßte sich auch eine *RZ-library* finden. Diese enthält die folgenden Module: *animated\_boolean*, *animated\_filename*, *mandelbrot*, *output\_a60*, *read\_any\_image*, *write\_a60\_yuv* und *write\_any\_image*. Die Quellen dieser Module stammen vom ftp-Server des Deutschen Klimarechenzentrums in Hamburg, wo der ftp-Server des IAC (International AVS Center) gespiegelt wird. Ich habe an diesen Quellen einige Änderungen vorgenommen.

Die Module werden im folgenden kurz beschrieben.

- *read\_any\_image/write\_any\_image*: Diese beiden Module importieren bzw. exportieren eine Reihe von gängigen Grafikformaten. So kann man zum Beispiel ein GIF-Image einlesen, unter AVS bearbeiten und als SGI-RGB-File abspeichern. Da eben die Möglichkeit besteht, die Grafiken unter AVS mit einer großen Menge von Hilfsmitteln zu bearbeiten, stellen diese beiden Module mehr als ein einfaches Konvertierungstool dar.

Beide Module basieren auf dem Programm *imconv* vom SDSC (San Diego Supercomputing Center). Leider sind für *imconv* nur binaries und keine Quelltexte erhältlich, so daß keine Änderungen möglich sind (dieses Programm hat Probleme beim Importieren von PostScript-Dateien).

Ich habe bei beiden Modulen einen *read*- bzw. *write*-Knopf eingefügt (s. Abbildung), der der Steuerung des Lese- bzw. Schreibprozesses dient. In ihrer ursprünglichen Fassung lasen bzw. schrieben die Module jedesmal die Daten, wenn sich an ihren Input-ports etwas änderte. Das ist bei AVS leider auch dann der Fall, wenn man keine Änderungen am eigentlichen Bild vornimmt, sondern beispielsweise nur das betreffende Fenster verschiebt. Dieses nicht immer wünschenswerte Verhalten wird nun umgangen, indem man den Lese- bzw. Schreibprozeß explizit startet. Tritt dabei ein Fehler auf, wird der Knopf automatisch zurückgesetzt. Ist der Vorgang erfolgreich, muß man den Knopf selbst zurücksetzen.

- *output\_a60*: Dieses Modul erlaubt es, AVS-Images direkt auf die Videoplatte des Abekas zu schreiben (das ist allerdings nur möglich, wenn der Benutzer auf dem Abekas schreibberechtigt ist). Man kann hierzu die Positi-

on im Timecode (hh:mm:ss:ff)<sup>1</sup> eingeben und die Aufzeichnung mit einem *record*-Knopf starten. Außerdem kann eingestellt werden, wie oft das Bild auf die Platte geschrieben werden soll. Ist das abgeschickte Bild kleiner als das D1<sup>2</sup>-Format (720 x 576 Pixel), wird es zentriert und mit schwarzem Rahmen versehen. Ist das Bild zu groß, werden ausgehend vom Mittelpunkt alle überstehenden Bereiche abgeschnitten. Ein Modul, das Bilder vergrößert und verkleinert und dabei die aspect ratio berücksichtigt, ist leider nicht vorhanden. Dieses Modul wurde an die europäische Fernsehnorm PAL angepaßt.

- *write\_a60\_yuv*:

Dieses Modul ähnelt dem vorherigen. Es schreibt die Bilder allerdings nicht per Remote-copy auf die Platte des Abekas, sondern im für den Abekas lesbaren YUV-Format in ein vorzugebendes Verzeichnis - vorausgesetzt, man hat ausreichend Platz (pro Bild exakt 829440 Byte) und ist in diesem Verzeichnis schreibberechtigt. Die Voreinstellung ist das */tmp*-Directory des Rechners, auf dem das Modul läuft (also eine SGI-Workstation) und sollte nicht genutzt werden. Besser ist ein Verzeichnis im eigenen Home-directory. Die gespeicherte Datei hat einen Namen der Form *prefix.number.suffix*, wobei *prefix* vom Benutzer frei definierbar ist (die Voreinstellung ist *a60\_image*), *number* eine dreistellige Zahl ist, die bei 000 beginnt und automatisch inkrementiert wird, und *suffix* als Kürzel *yuv* zur Klassifizierung der Datei dient.

Hinweis: Da der Abekas das PAL-Format zur Ausgabe nutzt, benötigt man 25 Bilder für eine Sekunde Video.

- *animated\_boolean/animated\_filename*: Diese beiden Module sollen das Speichern von Bildsequenzen erleichtern. Das Modul *animated\_boolean* liefert einen wechselnden booleschen Wert (also abwechselnd die Integer-Werte 0 und 1). Das kann nützlich sein, um die *read*- bzw. *write*-Knöpfe von *read\_any\_image/write\_any\_image* automatisch zu bedienen, da diese Knöpfe in AVS durch einen booleschen Datentyp repräsentiert werden. Das Modul *animated\_filename* erzeugt aus einem Präfix, einer laufenden Nummer und einem Suffix einen Dateinamen der Form *prefix.number.suffix*, wobei sich die Stelligkeit der Zahl *number* zwischen 1 und 10 variieren läßt. Das ist zum Einlesen einer Reihe aufeinanderfolgender Images nützlich, wie sie zum Beispiel mit *write\_a60\_yuv* erzeugt werden können.

- *mandelbrot*:

Dieses Modul erzeugt ein Bild der Mandelbrotmenge und dient hier nur der Unterhaltung.

<sup>1</sup> steht für hours:minutes:seconds:frames

<sup>2</sup> D1 ist das digitale Pendant zum analogen PAL

Genauere Beschreibungen für die Module sind als online-Dokumentationen unter AVS verfügbar.

**Erweiterungen**

Wie bereits erwähnt, stellen die bereits verfügbaren Module nur einen Grundstock dar. Wer sich für die Quellen interessiert, findet sie auf den SGI-Maschinen unter `/usr/avs/rzlib/src`. Dort sind auch die Makefiles zu finden, die man für eigene Zwecke

nur leicht zu modifizieren braucht. Man kann auch unter AVS den *Module Generator* benutzen, der automatisch die Grundgerüste für Quelltexte, Helpfiles etc. generiert. Die Erstellung von Helpfiles zu neuen Modulen ist anzuraten. Der Ansprechpartner für künftige Erweiterungen ist Herr Barz (Telefon: 2093-2363).

Ulrich Lissé

## Software zur 3D-Modellierung und Animation

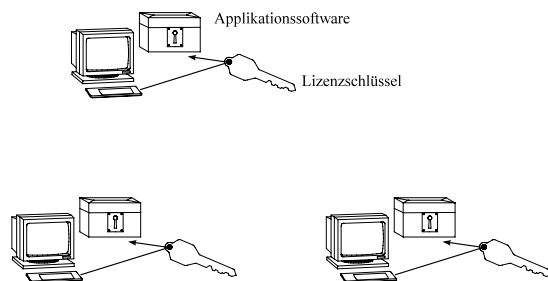
**Konfigurationsmöglichkeiten des Programmsystems der Firma Wavefront**

Zur 3D-Modellierung und Animation benutzen wir vorrangig Software der Firma Wavefront für Silicon Graphics Workstations. Die Komponenten: Data Vizualizer, Advanced Visualizer, Visualizer Paint und Composer hat das Rechenzentrum, wie schon in vorangegangenen Rechenzentrumsmitteilungen zu lesen war, als Einzellizenz erworben.

Die Firma Wavefront bietet für ihre Applikationssoftware drei Konfigurationsmöglichkeiten, die im folgenden kurz mit ihren Anwendungsempfehlungen, ihren Vor- und Nachteilen vorgestellt werden sollen. Zu jeder erworbenen Lizenz gehört ein Lizenzschlüssel, der dem Nutzer die Anwendung dieser Software erlaubt.

**Node-locked-Konfiguration (Abb.1)**

Darunter versteht man die lokale Installation der Applikationssoftware und der Lizenzschlüssel, das heißt, jede Workstation erhält eine eigene Kopie der Software und einen eigenen Lizenzschlüssel, der nur für diese entsprechende Workstation gilt.



Jede Workstation erhält eine Kopie der Applikationssoftware und einen eigenen Schlüssel.  
Abb.1

Diese Konfigurationsart empfiehlt sich für Einzellizenzen oder mehrere Workstations, die nicht durch ein Netz verbunden sind. Der Vorteil dieser Konfigu-

ration besteht im schnellen Zugriff zur Software. Die Nachteile sind der große Speicherplatzbedarf für jede installierte Softwarekopie, und bei neuen Softwareversionen muß auf jeder Workstation neu installiert werden.

**Softwareserver-Konfiguration (Abb. 2)**

Voraussetzung für diese Konfigurationsart ist die Netzanbindung der Workstations. Die Applikationssoftware wird auf einer Workstation im Netz - dem

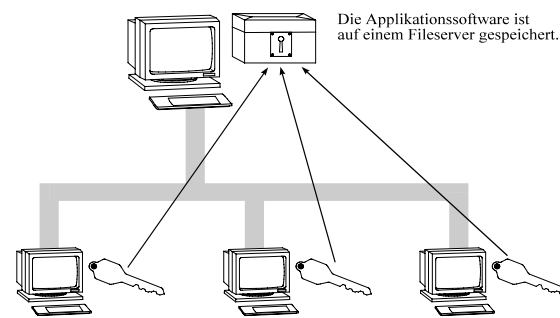


Abb. 2

Fileserver - geladen. Die Workstations in diesem Netz - die Clientworkstations -, die diese Software nutzen dürfen, erhalten einen Lizenzschlüssel, der lokal auf den entsprechenden Workstations gespeichert wird und nur für diese Workstations gilt. Damit sind sie privilegiert, die Applikationssoftware von der Fileserverplatte zu übertragen und abzuarbeiten.

Diese Form der Konfiguration bietet sich an, wenn ein Netz vorhanden ist und nicht genügend lokaler Speicherplatz zur Verfügung steht. Vorteile liegen im Sparen von Speicherplatz, und bei Softwareversionsaustausch wird nur auf einer Workstation neu installiert. Nachteil ist der langsamere Zugriff auf die Applikationssoftware über das Netz.