und im wesentlichen bieten sie alle das gleiche: Ein Bild in einem Rasterformat. Sie müssen dieses nun nach PostScript konvertieren. Dazu sollten Sie zuerst überlegen, ob es lohnt, das Bild zu vektorisieren. Wenn es nämlich nur aus dicken und dünnen Linien (oder Kreisen oder derartigem besteht), so können Sie durch das Vektorisieren des Bildes viel Speicherplatz sparen (dazu werde ich noch einiges ausführen). Ein Vektorisierungsprogramm, das aus den meisten Rasterbildformaten PostScript-Bilder im Vektorformat machen kann, stellt das HRZ zur Verfügung. Es heißt LS-Vektor und ist über Herrn Löw (Raum 109) zugänglich. Leider handelt es sich dabei nicht um ein PublicDomain- oder ShareWare-Programm, so daß wir es dem DVIPS-Paket nicht beilegen konnten. Wenn Sie also das Bild mit diesem Programm vektorisiert haben, dann haben Sie ein fertiges PostScript-Bild.

In vielen Fällen jedoch wird es nicht sinnvoll sein, das Bild zu vektorisieren, etwa dann, wenn es sich um ein (eingescanntes) Foto oder ein Bild mit mehreren ausgefüllten Flächen handelt. In diesem Falle muß das Bild ins PostScript-Rasterbildformat konvertiert werden. Dazu und zur Nachbearbeitung des Bildes gibt es im DVIPS-Paket zwei Konvertierungsprogramme: GraphicWorkshop und ALCHEMY. Bei beiden Programmen ist eine ellenlange Anleitung dabei, die Sie aber sicherlich nicht lesen wollen. Deshalb hier eine Kurzanleitung: Um ein beliebiges Rasterbild mit ALCHEMY nach PostScript zu konvertieren, geben Sie einfach folgende Befehlszeile ein:

`alchemy -e0 -b -8 -D300 -D300 -d1` <*Name des zu konvertierenden Bildes*>

Das sollte eigentlich immer klappen. Leider ist ALCHEMY ein ShareWare-Programm; das bedeutet, sie dürfen es nur auf begrenzte Zeit kostenfrei benutzen. Lesen Sie die entsprechenden Dateien, die bei ALCHEMY mitgeliefert werden. Auch unterliegt die nicht bezahlte ShareWare-Version (also die, die dem DVIPS-Paket beiliegt) von ALCHEMY einigen Einschränkungen[39].

Nach dieser Befehlszeile haben Sie also in der Regel Ihr Bild ins PostScript-Format konvertiert und es liegt als Encapsulated PostScript Image (abgekürzt EPI, verwendete Endungen `.EPI` oder `.EPS`) vor. Dieses können Sie nun weiterverarbeiten[40]. Die Vorzüge von ALCHEMY liegen vor allem in der Möglichkeit, TrueImage-Bilder verarbeiten zu können (das sind Bilder, in denen etwa 16 Millionen Farben gleichzeitig auftauchen können).

In der Regel werden Sie jedoch mit GraphicWorkshop (auch dieses ist ein ShareWare-Programm, beachten Sie die entsprechenden README-Dateien) vergleichbare Ergebnisse erzielen. Dieses Programm ist vollständig menüorientiert, so daß ich dazu keine weiteren Erklärungen abgeben will. Außderdem müssen Sie GraphicWorkshop an Ihre spezielle Grafikkarte anpassen (wenn Sie nicht gerade eine Standard-VGA-Karte haben) und dazu ohnehin die Anleitung lesen. Mit GraphicWorkshop haben Sie außer den Konvertierungsmöglichkeiten, die auch ALCHEMY bietet, noch einige zusätzliche Manipulationsmöglichkeiten für das Bild.

Rasterformat konvertieren können. Mit dem Programm GraphicWorkshop und dem kommerziellen Programm HiJaak können Sie jeweils nur eine Teilmenge davon bearbeiten, mit der man aber in aller Regel auskommt.

[39] Die bestehen in erster Linie darin, daß Sie mit ALCHEMY in der ShareWare-Version keine Rasterbilder verarbeiten können, die größer als 640×400 Punkte sind. Falls Sie häufig Bilder konvertieren, sollten Sie über die Anschaffung der Vollversion von ALCHEMY nachdenken. Eine Vollversion von ALCHEMY steht aber auch am HRZ zur Verfügung. Das Programm ALCHEMY ist nicht nur erheblich billiger als etwa das voll kommerzielle Programm HiJaak, es ist auch wesentlich leistungsfähiger und liefert bessere Ergebnisse.

[40] EPI-Dateien können Sie in der Regel nicht direkt an einen PostScript-Drucker schicken!

Es besteht zum Beispiel die Möglichkeit, das Bild zu vergrößern oder zu verkleinern, es zu invertieren (bei zweifarbigen Bildern manchmal sehr nützlich) oder zu drehen, zu spiegeln und noch so einiges mehr. Bei kontrastarmen Bildern (z. B. eingescannte Schwarz-Weiß-Fotos, die grau in grau erscheinen) erzielen Sie in aller Regel mit GraphicWorkshop bessere Ergebnisse als mit ALCHEMY, die noch besser werden, wenn Sie das Bild vor der Konvertierung um den Faktor 2 (das bedeutet bei GraphicWorkshop eine Vergrößerung auf 200% der Originalgröße) oder mehr vergrößern.

Beide Programme bieten die Möglichkeit, sich die zu konvertierenden Bilder vor der Konvertierung anzuschauen. Beim menüorientierten GraphicWorkshop ist das klar, bei ALCHEMY geht es mit

`alchemy -V` <*Name des Rasterbildes*>

wenn Sie eine Extended-VGA-Karte haben, oder mit

`alchemy -Vv` <*Name des Rasterbildes*>

wenn Sie nur eine Standard-VGA- oder EGA-Karte haben. Falls Sie noch nicht einmal eine EGA-Karte haben, so können Sie mit ALCHEMY zwar noch Bilder konvertieren, sie aber nicht mehr anschauen. GraphicWorkshop können Sie wahrscheinlich ohne mindestens EGA-Karte überhaupt nicht benutzen (ich habe es nie ausprobiert).

Was aber passiert mit Farbbildern? In der Regel können PostScript-Ausgabegeräte nur Graustufen darstellen, keine Farben. Es ist aber im PostScript-Sprachstandard eine Möglichkeit vorgesehen, auch Farben zu verarbeiten. Zur Zeit steht am HRZ noch kein Farb-PostScript-Drucker zur Verfügung, aber wenn Sie einen benutzen können, so spricht nichts dagegen, auch farbige Bilder einzubinden. Allerdings können Sie Farbbilder unter Erhaltung der Farbinformation nur mit ALCHEMY nach PostScript konvertieren (bei GraphicWorkshop geht bei der direkten Konvertierung, also ohne vorherige Graustufenrasterung, nach PostScript in jedem Falle die Farbinformation verloren). Hier werden Sie aber leicht an die Grenzen der ShareWare-Version von ALCHEMY stoßen. Das HRZ stellt allerdings noch ein kommerzielles Konvertierungsprogramm namens HiJaak zur Verfügung, das keine derartigen Einschränkungen hat, außerdem steht Ihnen ja (wie gesagt) eine Vollversion von ALCHEMY zur Verfügung. Falls Sie es benötigen, erkundigen Sie sich halt. Aber wie gesagt: In der Regel wird Ihnen kein farbenfreudiges Ausgabegerät zur Verfügung stehen. Was dann?

Dann muß man statt Farben Graustufen zur Darstellung benutzen. Da gibt es zum einen sehr raffinierte Wege zur Umwandlung von Farbbildern in reine Schwarz-Weiß-Pixelbilder (sogenannte Dithering-Verfahren) und zum anderen auch sehr pragmatische Verfahren. Welches besser ist, richtet sich nach dem zu konvertierenden Bild. Deshalb will ich für diesen Bereich jetzt ein wenig Erfahrungswissen rüberbringen.

Wenn Sie ein knallbuntes Bild haben, das also weniger eine Farbe in vielen verschiedenen Schattierungen enthält, sondern eher viele völlig verschiedene Farben, so ist es meist am besten, dieses Bild nicht weiter zu behandeln, sondern es einfach mit GraphicWorkshop in ein PostScript-Bild zu konvertieren. Die Farben im Bild sind in der Regel palettiert und GraphicWorkshop tauscht einfach die Farbpalette gegen eine Graustufenpalette aus. Das Erzeugen der verschiedenen Graustufen übernimmt in diesem Fall der PostScript-Drucker. Falls Sie auf diese Weise z. B. ein rein blaues Bild konvertieren (also ein Bild das zum Beispiel hellblau, mittelblau, ozeanblau, dunkelblau, marineblau, ... kurzum viele verschiedene Blautöne enthält), so werden Sie das Bild in der PostScript-Darstellung unter Umständen hervorragend wiedergegeben oder auch stark verfälscht finden. Aus dem hellsten Blauton wurde dann nämlich weiß, aus dem dunkelsten Blauton wurde (in der Regel) reines Schwarz. Oder umgekehrt. Oder mittelblau wurde schwarz, himmelblau wurde weiß. Dieses Vorgehen streckt natürlich den Kontrastbereich, liefert also kontraststärkere Bilder, wird aber nicht immer erwünscht sein. Falls also dieses Verfahren „in die Hose geht" und die Zuordnung der Graustufen nicht zu den originalen Farben paßt, dann muß das Bild vor der Konvertierung „gedithert" werden, das bedeutet, man generiert die Graustufen selber und überläßt die Zuordnung von Faben zu Graustufen nicht dem Zufall oder dem PostScript-Ausgabegerät. Das geht sowohl mit GraphicWorkshop (es wird eine Vielzahl verschiedener Dithering-Verfahren angeboten, am besten fährt man in der Regel mit dem bidirektionalen Floyd-Steinberg-Verfahren) als auch mit ALCHEMY. Nach meiner eigenen Erfahrung und meiner ganz persönlichen Meinung erzielt man hier mit ALCHEMY bessere Ergebnisse.

In der obigen Kommandozeile für den ALCHEMY-Aufruf wird folgendes gemacht: Zunächst soll das Bild in ein PostScript-Bild konvertiert werden (Parameter `-e0`), dann soll es ein schwarz-weißes PostScript-Bild werden (Parameter `-b`, falls Sie also die Farbinformation erhalten möchten, müßten Sie diesen Parameter einfach weglassen. Allerdings können Sie das Bild dann mit den am HRZ zur Verfügung stehenden PostScript-Druckern nicht mehr ausdrucken.), die Graustufen-Information soll in einer Palette abgelegt werden (Parameter `-8`, das heißt also, das PostScript-Ausgabegerät bekommt die Palette mitgeliefert und wird nicht seine eigene Standard-Graustufenpalette verwenden, was bewirkt, daß die Bilder wirklich auf allen PostScript-Ausgabegeräten gleich aussehen), das Ausgabegerät hat eine Punktdichte von 300 dpi sowohl in horizontaler als auch in vertikaler Richtung (Parameter `-D300 -D300`) und das Bild soll, wenn nötig, nach dem bidirektionalen Floyd-Steinberg-Verfahren gedithert werden (Parameter `-d1`). Andere Dithering-Verfahren stünden Ihnen mit `-d2` oder `-d3` zur Verfügung. Wenn Sie mehr wissen möchten, sei Ihnen die Lektüre der ALCHEMY-Anleitung ans Herz gelegt.

Ansonsten: Fröhliches Konvertieren. Viel falsch machen können Sie eigentlich nicht. Außerdem haben Sie ja die Möglichkeit, sich die konvertierten Ergebnisse vor dem Ausdruck mit GhostScript am Bildschirm zu betrachten.

9.2 Sie haben ein PostScript-Bild

Wunderbar, Sie haben also auf irgendeine Weise ein PostScript-Bild bekommen. In der Regel werden Sie dann ein Encapsulated PostScript Image (`.EPI` oder `.EPS`) haben. PostScript-Dateien (und somit auch EPI-Dateien) können Sie sich mit jedem Editor ansehen[41]. Der Anfang einer solchen Datei sieht dann z. B. so aus:

```
%!PS-Adobe-2.0 EPSF-2.0
%%BoundingBox: 0 0 122 95
%%Title: ( FAMILY.EPI )
```

[41] Sie können aber nur vollwertige PostScript-Dateien (`.PS`) direkt an einen Drucker schicken; mit EPI-Dateien geht das in der Regel nicht! Die müssen erst eingebunden werden.

```
%%Creator: Graphic Workshop 4.8 from Alchemy Mindworks Inc.
%%CreationDate: Sat Nov 02 17:52:16 1991
%%EndComments
/width 512 def
/height 400 def
/pixwidth 122.879017 def
/pixheight 95.999232 def
/picstr width string def
/dopic {
gsave width height 8
[width 0 0 height neg 0 height]
{currentfile picstr readhexstring pop}
image grestore } def
0 0 translate pixwidth pixheight scale
dopic
152323231512050A0E0E080505050505080808080E121C1C1C15151515150A0A0A
080208080808080802050502020202020505050A0A1212150A0E120E1C1C1C1C23
232B2B231C150E0A0A0E0E15151C2323232B2B2B2B2B2B23232315151C1C1C1C
15151C1512151512151C1C232B231C231C1C232323232B2B324A534A3A2B3A3A
3A323A3A3A3A3A323A322B32322B2B322B232B2B232B2B2B2B322B3A3A323232
...
```

Ich vermute mal, das wirkt eher abschreckend. Ist auch gar nicht schlimm. Wichtig für Sie ist im Prinzip nur die zweite Zeile. Hier steht der sogenannte Bounding-Box-Kommentar. Dieser sagt aus, wie groß das nachfolgende Bild wird und ist somit von elementarer Wichtigkeit. Wenn in Ihrem PostScript-Bild aus irgendwelchen Gründen kein Bounding-Box-Kommentar enthalten ist oder dieser nicht genau so aussieht wie oben (zunächst der Text `%%BoundingBox:` und dann vier Zahlen und das ganze in einer eigenen Zeile, in der sonst nichts steht), so sorgen Sie dafür, daß sich das ändert. Wo in der Datei diese Bounding-Box-Angabe steht, ist völlig egal. Es ist allerdings vorteilhaft, sie ganz an den Anfang **(aber nicht in die erste Zeile! Dort muß auf jeden Fall die Zeichenfolge `%!` erscheinen)** zu setzen[42], dann kann das Bild nämlich schneller verarbeitet werden. Ansonsten müßte zunächst die gesamte Datei nach dieser Zeile durchsucht werden und das kann bei größeren Dateien dann schon mal einige Minuten dauern. Wie sie die korrekte Bounding-Box eines PostScript-Bildes, das keine solche Angabe enthält, herausfinden können, ist im Kapitel über GhostScript beschrieben.

Also, nun haben Sie ein PostScript-Bild und das steckt in der Datei `FAMILY.EPI`. Was dann? Wir kommen nun also zur Benutzung des

Styles `PSFIG`

Zunächst einmal müssen Sie natürlich den Style `PSFIG` in Ihre Dokumentenstiloptionen aufnehmen, also

[42] Das mag sich unglaublich kompliziert anhören und ist es in gewissem Sinne auch. Falls Ihnen in der Tat mal eine EPI-Datei in die Hände kommt, die nicht diesen Konventionen entspricht, dann bitte ich darum, mir diese Datei zukommen zu lassen. Ich werde sie in meine Raritäten-Sammlung aufnehmen. Ernsthaft gesagt: Ich kann mir kaum vorstellen, wie aus irgendeiner Bildquelle durch Konvertierung oder Einscannen oder ähnliches eine nicht den Normen entsprechende EPI-Datei entstehen sollte.

```
\documentstyle[11pt,german,psfig]{article}
```

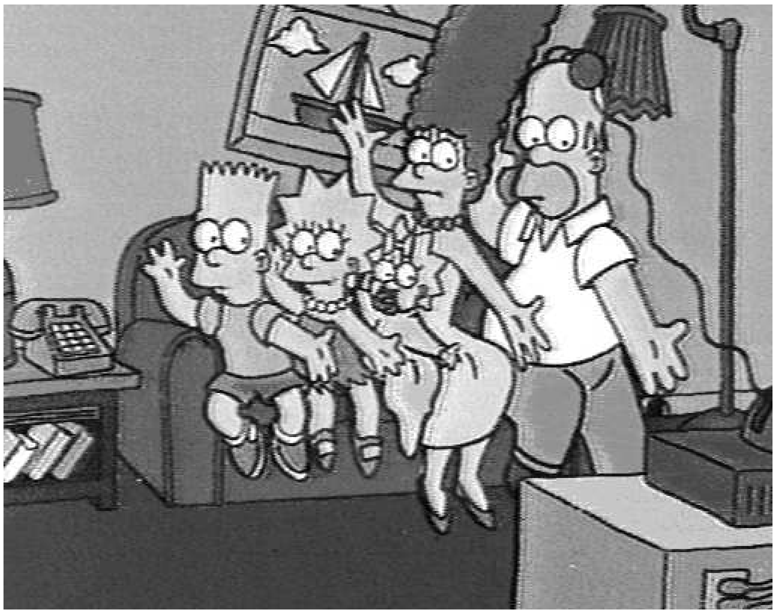
oder so ähnlich. **Achten darauf, daß Sie `PSFIG.STY` als letzte Stiloption angeben**. Nun haben Sie zum Einbinden der Bilder einen zusätzlichen Befehl, nämlich

`\psfig{figure=`<*Dateiname*>`,height=`<*Höhe*>`,width=`<*Breite*>`}`

und das war es im Prinzip auch schon. **Achten Sie darauf, daß innerhalb des `\psfig`-Aufrufs keinerlei Leerzeichen auftreten**. Das führt sonst zu Fehlern, die Sie stundenlang mit wachsender Begeisterung suchen werden (fragen Sie aber bitte nicht, woher diese Fehler kommen. Eigentlich sollte es egal sein, ob Sie eine Länge als 8 cm oder 8cm schreiben). Um also in den Text jetzt das in der Datei `FAMILY.EPI` (deren Anfang oben zu sehen war) enthaltene Bild einzubinden, gebe ich die Kommandos

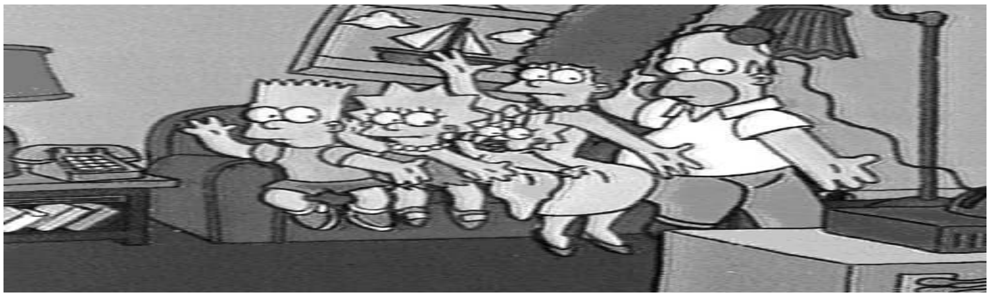
```
\centerline{\psfig{figure=FAMILY.EPI,height=8cm}}
```

und schon...



Wenn Sie nachmessen, werden Sie feststellen, daß das Bild in der Tat 8 cm hoch ist. Wenn keine Höhe oder Breite angegeben wird, so wird das Bild in seiner natürlichen Höhe und Breite dargestellt. Das ist jedoch selten erwünscht. Wenn Sie nur eine Höhe oder nur eine Breite angeben, so wird das Bild in eben dieser Breite oder Höhe dargestellt ohne allerdings dabei das gegebene Verhältnis Höhe zur Breite zu ändern (das Bild wird also nicht verzerrt). Das wird in aller Regel auch erwünscht sein. Es kann aber auch sein, daß Sie das Bild mit fest vorgegebener Höhe **und** Breite einbinden wollen. Dann wird das Bild eben entsprechend verzerrt. Ein Beispiel:

```
\centerline{\psfig{figure=FAMILY.EPI,height=3.8cm,width=13cm}}
```



Sie können die Höhen- und Breitenangaben auch in jeder anderen für TEX verdaulichen Einheit machen, also in Inch oder Punkt (`in` oder `pt`) oder sonst etwas. Am vorteilhaftesten wird es im allgemeinen sein, das Bild innerhalb einer LATEX-Abbildung (`figure`-Umgebung) darzustellen, also etwa mit

```
\begin{figure}[htp]
\centerline{\psfig{figure=FAMILY.EPI,height=4cm,width=12cm}}
\caption{Die Simpsons, die gesamte Familie}
\end{figure}
```

was unter anderem den Vorteil hat, daß dann automatisch Bildunterschriften erzeugt werden können, daß die Abbildungen automatisch durchnumeriert werden können, und sie außerdem nach geltenden typographischen Regeln plaziert werden, und nicht kunterbunt, wie ich das hier mache. Sie sollten sich meine Art der Bildplazierung nicht zum Vorbild nehmen, ich habe mehr didaktische Gründe, es so zu machen, wie Sie es hier vorfinden. Normalerweise werden die Bilder „schöner" plaziert, wenn Sie die Plazierung von der LATEX-`figure`-Umgebung vornehmen lassen[43].

Das scheinbar obligatorische `\centerline` kann natürlich auch weggelassen werden, dann landet das Bild halt am linken Rand. Mit entsprechenden LATEX-Befehlen kann man das Bild ebenso positionieren wie jede andere Box auch, denn der Befehl `\psfig` macht nichts anderes als eine Box. Wie man die Bilder nicht nur verzerren, sondern auch drehen kann, dazu siehe im Kapitel über DVIPS-Extras.

Es mag auch sinnvoll sein, die Bilder in einem anderen Verzeichnis aufzubewahren als die Texte. Wie aber teilt man dies mit? In einem solchen Falle müssen Sie nicht nur den Dateinamen, sondern den vollständigen Pfadnamen der Bilddatei angeben, also etwa

```
\psfig{figure=D:/BILDER/FAMILY.EPI,height=4cm}
```

Beachten Sie bitte, daß die Verzeichnisse nicht durch Backslashes (`\`), sondern durch den „normalen" Schrägstrich (`/`) getrennt werden. Ansonsten geben Sie den Pfadnamen ein wie gewohnt. Wenn Sie also eine Datei, in der mit PSFIG Bilder eingebunden wurden, TEXen und ganz seltsame Dinge passieren (z. B. erscheint plötzlich die Eingabeaufforderung `\line@in=` am Bildschirm oder Sie bekommen den Fehler `FATAL ERROR: No bb supplied`), versuchen

[43] Hier sollte ich noch auf den Style `FLOATFIG.STY` hinweisen, der ebenfalls mit dem DVIPS-Paket ausgegeben wird. Er erlaubt es, Texte um Abbildungen herumfließen zu lassen. Zu dem Style `FLOATFIG.STY` gehört eine hervorragende Anleitung, die Sie sich am besten selber mal anschauen. Da sie nur wenige Seiten lang ist, sehe ich nicht ein, sie hier noch einmal zusammenzufassen.

Sie es mal mit der Angabe der vollständigen Pfadnamen für die Bilder und wenn das auch nicht hilft, dann kontrollieren Sie, ob alle Bilder korrekte Bounding-Box-Kommentare enthalten.

Übrigens: Die EPI-Dateien (also die Bilder) werden sowohl benötigt, wenn Sie Ihre TEX-Datei mit TEX in eine DVI-Datei übersetzen, als auch, wenn Sie Ihre dann entstandene DVI-Datei in eine PostScript-Datei übersetzen! Erst, wenn Sie die fertige PostScript-Datei haben, die Ihnen DVIPS liefert, werden die Bilder nicht mehr gebraucht, da sie (physikalisch) in die PostScript-Datei kopiert wurden. Das ist auch der Grund dafür, daß DVIPS bei der Bildeinbindung ziemlich stark auf der Platte „herumackert". Derartiges Verhalten muß ja nicht immer auf einen Computervirus hindeuten, der gerade Ihre Platte formatiert (könnte es aber, man weiß ja nie...). Wegen des Anlegens einer echten Kopie der Bildinformationen in der PostScript-Datei müssen Sie auch für jedes eingebundene Bild den doppelten Speicherplatz einkalkulieren: Einmal ist das Bild sowieso da, und dann wird es nochmal in die entstehende PostScript-Datei hineinkopiert; und wenn das Bild dann an späterer Stelle noch einmal auftaucht, wird es auch noch einmal kopiert! Wenn Sie also Speicherplatzprobleme bekommen, sollten Sie eventuell die entstehende PostScript-Datei nicht zwischenspeichern, sondern direkt auf einen PostScript-Drucker schicken. Das ist aber auch am HRZ nur an den Geräten möglich, die direkt an einen PostScript-Drucker angeschlossen sind.

Hier gehört nun noch ein Einschub rein: Wenn Sie mit einem normalen Editor unter MS-DOS EPI-Dateien (oder ganz allgemein PostScript-Dateien) anschauen, so werden Sie in manchen Fällen feststellen, daß die Editoren die Zeilen nicht als solche erkennen und die Datei unter Umständen so behandeln, als bestünde sie nur aus einer Zeile. Das hat einen einfachen Grund: Unter MS-DOS gilt die Kombination der Zeichen CR (=Carriage Return) und LF (=Linefeed) als Zeilenende[44], unter vielen anderen Betriebssystemen (allen Betriebssystemen aus der UNIX-Welt, wie etwa Sun OS, Amiga-DOS, SINIX und viele andere) wird diese Funktion jedoch vom Zeichen LF (Linefeed, Code 11) alleine erfüllt. Eine PostScript-Datei also, die einem UNIX-System entspringt oder von einem aus der UNIX-Welt stammenden Programm erstellt wurde, (dazu gehört nun tragischerweise auch ALCHEMY) hat LF als Zeilen-Ende-Zeichen. Das ist nicht weiter schlimm und macht weder dem Programm DVIPS noch dem Style `PSFIG` (bzw. emTEX) etwas aus. Aber wenn Sie die Datei von Hand nachbearbeiten wollen, müssen Sie die Zeilen-Ende-Zeichen in der Regel austauschen (es sei denn, Sie haben ein fähiges Editor-Programm). Dazu ist dem DVIPS-Paket das Programm `LFCR.EXE` mitgegeben. Wenn Sie es ohne Parameter aufrufen, erklärt es Ihnen, wie man es bedient.

9.2.1 Und noch einige gute Ratschläge

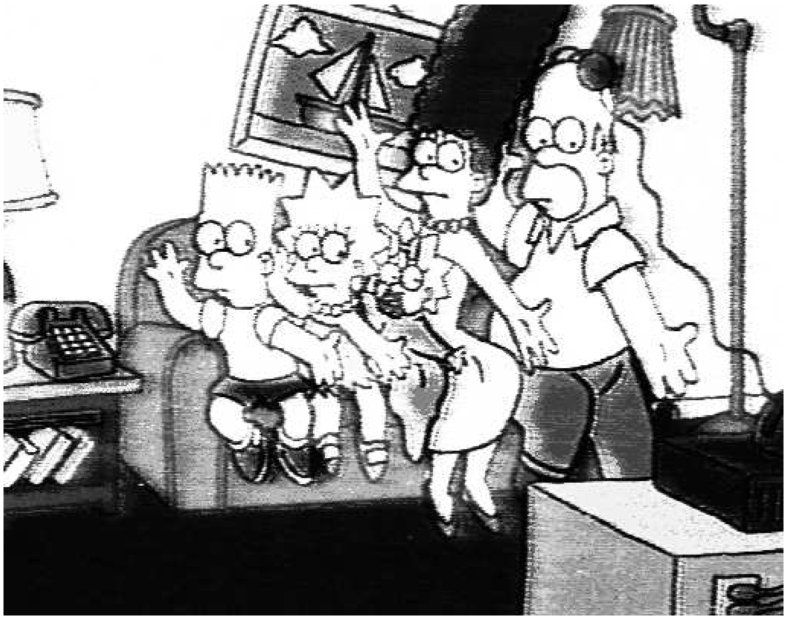
Ich will nun etwas genauer auf das Beispielbild eingehen. Wie Sie sicher schon erraten haben, handelt es sich um ein Bild aus der Fernsehserie *„Die Simpsons"*. Ich bin dabei von einem Rasterbild im GIF-Format[45] (512×400 Punkte, 256 Farben) ausgegangen, das ich von irgendwoher bekommen habe[46]. Dieses habe ich dann mit ALCHEMY genau mit der weiter oben

[44] Einige Editoren begnügen sich allerdings auch mit dem Zeichen CR als Zeilenende.

[45] Das GIF-Format ist im PC-Bereich das am häufigsten benutzte Format und somit ein Quasi-Standard. Im professionellen Bereich aber wird dem GIF-Format allgemein das TIF-Format vorgezogen.

[46] Um genau zu sein: Ich habe es via FTP von einem kalifornischen Fileserver geholt. Über FTP und Fileserver können Sie Informationen bei fast allen HRZ-Mitarbeitern bekommen. Es gibt überall in der Welt Fileserver, die Unmengen von GIF-Bildern bereitstellen, unter anderem möchte ich für Fans den Server `jumbo.hartford.edu` vorstellen, der 450 eingescannte Fernsehbilder (insgesamt 67500 KB) aus allen Folgen der Serie *„Startrek -*

angeführten Befehlszeile ins PostScript-Format konvertiert. Wenn man dagegen zur Konvertierung GraphicWorkshop verwendet (ohne Dithering, ohne jegliche sonstige Behandlung), dann sieht das Ergebnis so aus:



Sie sehen deutliche Unterschiede. Es ist wohl auch eine Frage des Geschmacks, welches Konvertierungsergebnis einem hier mehr zusagt. Dieses bisher verwendete Bild ist (wie Sie leicht nachvollziehen können) **nicht** zur Vektorisierung geeignet.

PostScript-Rasterbilder haben allerdings neben allen Vorteilen auch eine sehr unangenehme Eigenschaft: Sie fressen unglaublich viel Speicherplatz. Das in den bisherigen Beispielen zugrunde liegende GIF-Bild beanspruchte 140 KB Speicher. Das daraus entstandene EPI-Bild fraß aber ca. 450 KB Speicher, mehr als 3 mal so viel (und dabei hatte es noch wesentlich weniger Farben)! Und dieses Verhältnis ist noch mäßig. Um die Problematik zu verdeutlichen, will ich ein weiteres Bildbeispiel anführen:

---

the next generation" bereitstellt. Wenn man lange genug sucht, findet man vielleicht auch irgendwo sinnvolle GIF-Bilder (ich meine solche, die man selber verwenden kann)...