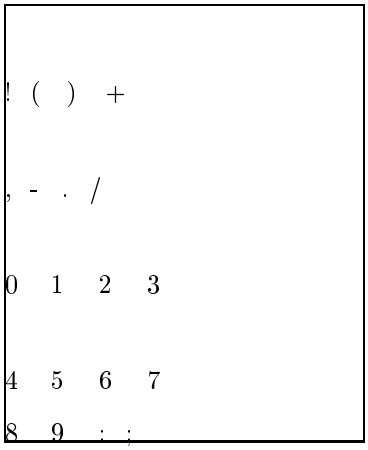
Benutzerhandbuch BM2FONT

Heinrich-Heine-Universität
Düsseldorf
Universitätsrechenzentrum
Friedhelm Sowa

Version 1.0 — 1. Juli 1991



Integration
von
Grafiken
und
Bildern
in TEX–Dokumente

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einführung

Die Integration von Zeichnungen und Bildern in TEX– oder LATEX– Dokumente ist bisher auf vielen verschiedenen Wegen realisiert worden. Der Mangel der meisten dieser Lösungen liegt in der Hauptsache darin begründet, daß sie sehr stark auf spezielle Treiberprogramme abstellten. Es mutet fast anachronistisch an, wenn mehr als zehn Jahre nach der Freigabe von TEX als „public domain“ Textsatzsystem noch keine Lösung der Grafikintegration gefunden wurde, die auch von frei verfügbaren Treibern bewältigt werden kann.

Zweifellos ist dieses Problem mit der wachsenden Zahl leicht zu handhabender und auch komfortabler Grafiksysteme speziell im Bereich der Personal Computer immer drängender geworden. Dazu kommt noch die Verfügbarkeit von optischen Erfassungssystemen wie Scanner, Kameras oder Videoscanner, die teilweise hochqualitative Bilder liefern. Es wurde im Laufe der Zeit immer weniger einsehbar, daß es keine zufriedenstellende Schnittstelle zu einem der qualitativ besten und auf nahezu allen Rechnersystemen verfügbaren Textsatzsysteme geben soll.

Somit scheint der Weg für die Lösung des Problems der Grafikintegration in TEX klar zu sein:

1. Nimm ein Grafiksystem und erzeuge ein Bild!
2. Mache dieses Bild TEX verständlich und übersetze es gleichzeitig in eine Sprache, die jeder Treiber versteht!
3. Setze das Bild an die passende Stelle im Text!
4. TEX setzt das Dokument.
5. Ein Treiber visualisiert das Dokument.

Der zweite Schritt wird mit BM2FONT durchgeführt, womit die bisher in dieser Form fehlende Schnittstelle zu TEX hergestellt ist. Obwohl diese Lücke auf einem PC gefüllt wurde, bedeutet dies keine Einschränkung der Geräteunabhängigkeit. Sie bleibt erhalten, da die Ausgaben von BM2FONT auf jedem

Rechner unter jedem Betriebssystem, für das eine Implementation von TEX existiert, verstanden werden.

2 Bildquellen

Der Inhalt einer Grafikdatei kann entweder in Form von Vektoren oder als Raster vorliegen. Hinsichtlich der Ausgabegeräte stellt die Vektorgrafik letztendlich eine Phase der Metamorphose im Zyklus der Entstehung einer Abbildung dar. Wie überall in der elektronischen Datenverarbeitung wird auch bei Grafiken alles auf zwei Zustände zurückgeführt: Entweder ist ein Bit gesetzt und hat den Wert 1 oder es ist nicht gesetzt und hat den Wert 0. Auf dem Papier erscheinen entweder schwarze oder weiße Stellen.

Diese Erkenntnis ist absolut nicht neu und hat zu einer Vielzahl von Konvertierungsprogrammen geführt, die Vektorgrafiken in Rastergrafiken umwandeln. Teilweise erlauben auch Grafiksysteme, die mit Vektorgrafik arbeiten, die Ausgabe des Bildes als Rastergrafik.

In der Praxis existieren zudem viele Grafiksysteme, die mit Rastergrafik arbeiten. Dies ist gerade in der PC-Welt sehr häufig der Fall. Es haben sich Quasi-Standards für die Speicherung von Rastergrafiken etabliert, die von vielen Programmsystemen verarbeitbar sind.

Bei der Frage, welche Form der Grafikspeicherung von BM2FONT als Grundlage angenommen werden sollte, ging kein Weg an der Rastergrafik als Antwort vorbei. Damit kann der weitaus größte Teil potentieller Anwendungen abgedeckt werden. Zudem ist allen Arten von Abbildungen, seien es z. B. technische Zeichnungen oder auch Fotos, der Weg in ein TEX–Dokument offen.

BM2FONT verarbeitet die nachfolgend aufgeführten Bitmap–Formate. Am Suffix der Eingabedatei wird das Format erkannt und – soweit vorhanden – die Identifikation der Datei entsprechend den verlangten Konventionen geprüft. Bei Abweichungen erfolgt ein entsprechender Hinweis. Sofern es möglich ist, wird auch bei Abweichungen die Konvertierung fortgesetzt, unter Umständen kann es aber dann doch noch zu unerwarteten Ergebnissen kommen. Das ist

sicherlich auch darin begründet, daß die Formate teilweise nicht allumfassend offengelegt sind, und so bei Systemen, die diese Formate verwenden, fehlerhafte Informationen erzeugt werden. BM2FONT unterstützt jedoch alle zweifelsfreien Features der Formate, es taucht kein Hinweis wie „keep your fingers crossed“ auf.

2.1 PCX-Dateien

Der PCX–Standard wurde von der Firma ZSOFT zur Speicherung von Bildschirmgrafiken entwickelt. Er erlaubt die Verwendung von bis zu 256 Farben (bis zu Version 3.0 nur 16 Farben) in einem Bild und bietet eine effiziente Form der Speicherung der Bildinformation. Es wird ein RunLengthEncoding–Schema verwendet, das den Speicherbedarf für ein Bild deutlich reduziert. Diese Form der Speicherung wird auch von einigen Capture–Programmen, die das Abspeichern von Bildschirminhalten vornehmen, verwendet.

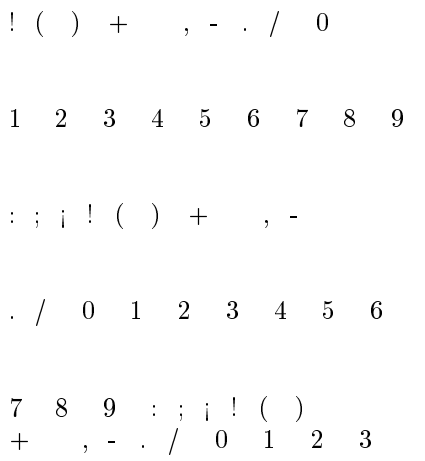


Abbildung 1: Beispiel einer PCX-Datei mit 16 Graustufen, ausgegeben mit 15 Graustufen

Neben einigen Scanner-Programmen unterstützen Zeichenprogramme wie beispielsweise *Paintbrush*, PC Paintbrush+, DELUXE PAINT II oder Publishers Paintbrush das PCX-Format.

2.2 GIF-Dateien

Der GIF-Standard wurde von der Firma CompuServe 1987 eingeführt. Das Graphics Interchange Format ist hardwareunabhängig und dient dem Austausch von Grafikdaten innerhalb vernetzter Systeme. Es benutzt zur Komprimierung den LZW-Algorithmus, der eine sehr hohe Effizienz hat.

Von BM2FONT werden fragmentierte Bilder in GIF-Dateien nicht unterstützt, obwohl der Standard die Speicherung mehrerer Bilder in einer Datei zuläßt. Der Grund ist darin zu sehen, daß in der Regel Bilder einzeln und im Text unterschiedlich verteilt gesetzt werden. Bei mehreren Bildern aus einer Datei müßte eine generische Namensvergabe für die Bilder erfolgen, was zu Kollisionen führen könnte.

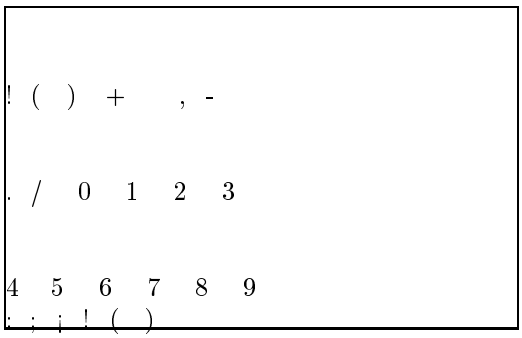


Abbildung 2: Beispiel einer GIF-Datei mit 256 Farben, ausgegeben mit 16 Graustufen und Separation der Halbtonpunkte

Obwohl laut Beschreibung des Standards eine Information über das Verhältnis zwischen der Breite und der Höhe eines Bildpunktes vorhanden ist, wird sie in den meisten Fällen nicht gegeben. Anhand der horizontalen und vertikalen Auflösung (Bits pro Zoll) des Bildes kann jedoch ein EGA-Bild erkannt werden.

BM2FONT nimmt automatisch eine Streckung des Bildes in vertikaler Richtung vor, so daß Eindrücke, wie sie Quasimodo vermittelt, vermieden werden. Sollte in dieser Hinsicht jedoch einmal zuviel des Guten getan worden sein, kann die Streckung im Einzelfall durch entsprechende Parametereinstellung verhindert werden.

2.3 BMP-Dateien

Zusammen mit Windows 3.0 hat sich der BMP-Standard (device-independent bitmap format) verbreitet. Es wird neben der bitmap-Speicherung, die zwar speicheraufwendig aber doch zulässig ist, nach RLE-Komprimierungen für 4Bit- und 8Bit-Pixel unterschieden.

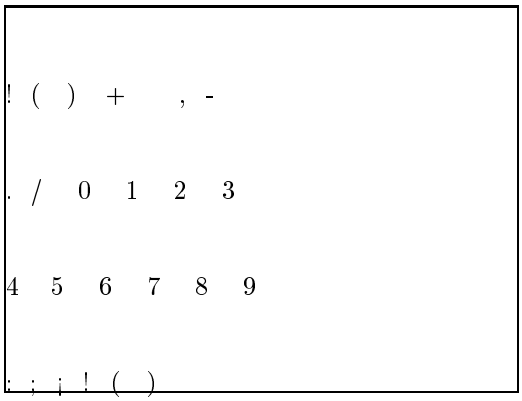


Abbildung 3: Beispiel einer BMP-Datei mit 16 Farben, ausgegeben mit 16 Graustufen und Separation der Halbtonpunkte

BM2FONT unterstützt die RLE-Komprimierung von BMP-Grafiken in der vorliegenden Version nicht. Der Grund ist schlicht und einfach darin zu sehen, daß zum Zeitpunkt der Auslieferung keine Testmöglichkeit für komprimierte Grafikdateien gegeben war. Dies ließ auch den Schluß zu, daß die Grafikprogramme in der Standardversion von Windows 3.0 für PCs diese Komprimierung nicht unterstützen. Insofern scheint an dieser Stelle keine sehr große Lücke zu existieren.

2.4 IFF- oder LBM-Dateien

Der Standard des Interchange File Format wurde von der Firma Electronic Arts eingeführt. Das Format wurde vorwiegend auf Amiga Rechnern benutzt, fand jedoch nach Einführung der Programme DeLuxe Paint und DeLuxe Paint II auch in der PC-Welt weite Verbreitung.

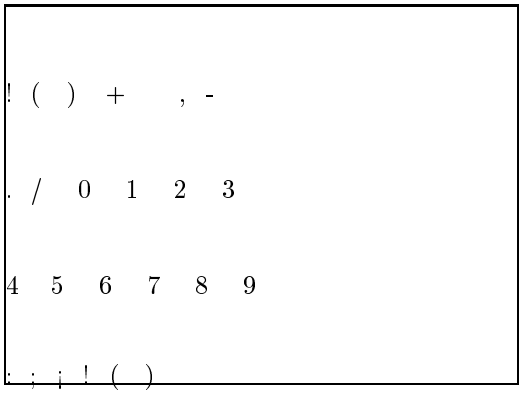


Abbildung 4: Beispiel einer IFF-Datei mit 256 Farben, ausgegeben mit 16 Graustufen und Separation der Halbtonpunkte

Die IFF-Dateien werden üblicherweise durch die Extension `.lbm` gekennzeichnet, BM2FONT ist aber tolerant genug, auch eine Datei mit dem Namen `newtut.iff` zu verarbeiten.

2.5 TIFF-Dateien

Das Tag Image File Format der Firma Aldus ist eines der am weitesten verbreiteten Standards zur Speicherung von Grafiken. Es wurde von vielen verschiedenen Herstellern akzeptiert und ist so auch auf verschiedenen Rechnersystemen anzutreffen. Auch Hersteller von Scannern unterstützen TIFF als Form der Speicherung von Bilddaten, da es schwarz/weiß-, Grau- und Farbbilder gleichermaßen erlaubt.

An dieser Stelle soll der Hinweis auf die Qualität von Halbtonbildern, die mit einem Scanner erzielt werden kann, nicht fehlen.

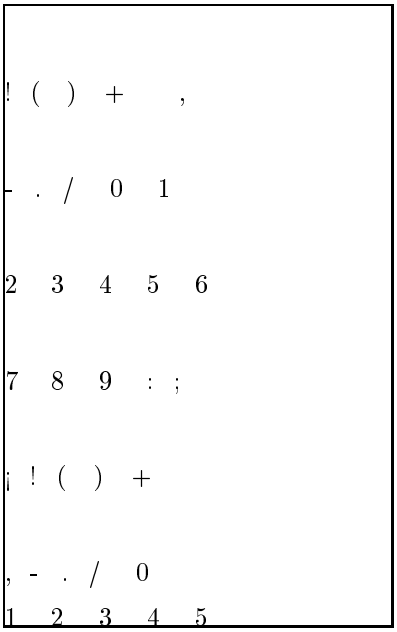


Abbildung 5: Beispiel einer TIFF-Datei mit 16 Graustufen, ausgegeben mit 16 Graustufen und Separation der Halbtonpunkte

Nicht die Auflösung, mit der ein Bild gescannt werden kann, ist entscheidend. Vielmehr die Anzahl der möglichen Farbabstufungen ist der wichtige Faktor bei der Qualität eines Bildes.

2.6 IMG-Dateien

Einige Grafik-Programme benutzen zur Speicherung von Bildern das GEM Image File Format. Dieses Format komprimiert den Inhalt des Bildschirmspei-

chers mit Hilfe unterschiedlicher Methoden. Es werden sowohl RLE-Encoding als auch Bitstreams, Muster und Wiederholfolgen benutzt. Das folgende Beispiel ist mit dem Programm SIRGraph entstanden, das auf mehrere Rechnersystem portiert worden ist.

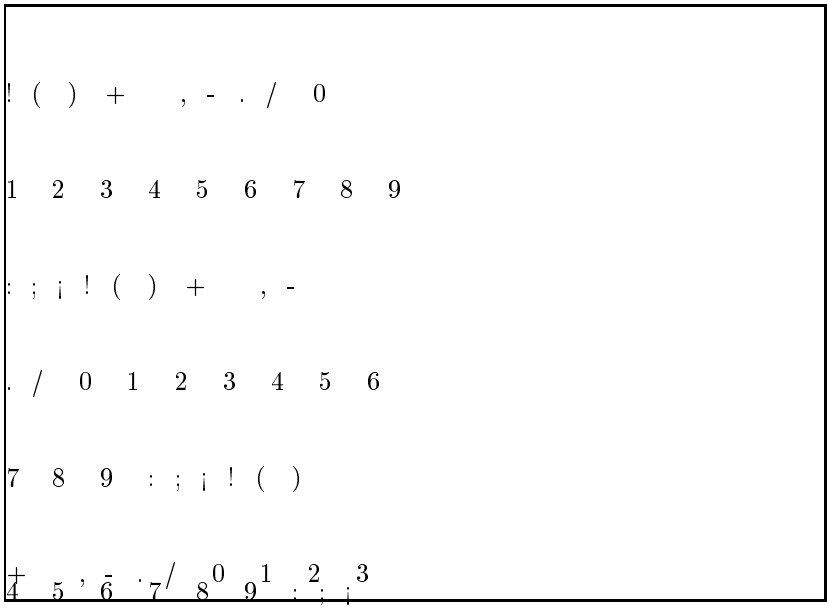


Abbildung 6: Beispiel einer IMG-Datei mit 16 Farben, ausgegeben mit 16 Graustufen, keine Fehlerverteilung

Anzumerken ist bei diesem Bild, daß die Grafiken durchaus auch auf Großrechnern entstanden sein können.

2.7 CUT-Dateien

Das CUT-Format wird zur Bildspeicherung unter anderem vom Videosystem ImagePro (Dr. Halo) verwendet. Es ist ein reines RunLengthEncoding Format. Je nach Hardwareausstattung kann es erforderlich sein, das von der Videoka-

mera gelieferte Bild in der Höhe zu reduzieren. BM2FONT geht von einem x-y-Verhältnis der gelieferten Graupixel von 5 zu 7 aus. Entsprechend wird das Bild bearbeitet. Durch Parametrisierung kann die Reduzierung abgeschaltet werden.

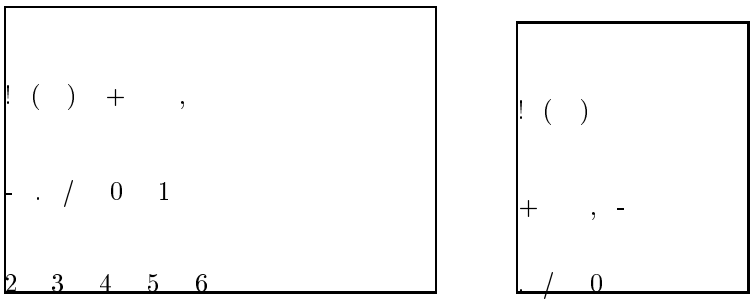


Abbildung 7: Beispiel zu CUT-Dateien mit 256 Graustufen, ausgegeben mit 36 Graustufen

2.8 Bitmaps

Natürlich war der Weg offenzuhalten für Bilder, die keiner dieser bislang aufgeführten Konventionen genügen. In diesem Fall, wenn auch alle verfügbaren Konvertierungsprogramme versagen, kann immer noch die Ausgabe einer Grafik als reine Bitmap in eine Datei erfolgen. Dies setzt allerdings spezielle Software voraus, die unter Umständen selbst erstellt werden muß.

BM2FONT kann eine Bitmap ohne irgendwelche begleitenden Informationen nicht am Suffix des Dateinamens erkennen. Durch Angabe der Breite des Bildes in Bytes mit Hilfe des Parameters l, wie z.B. `-l320` geht BM2FONT davon aus, daß es sich um eine einfache Bitmap handelt.

Das folgende Bild wurde mit `bm2font venus.bit -l320 -gy -x4 -u2 -b2` generiert. Es lag ursprünglich als Datei `venus.gif` vor, jedoch war der rechte Rand des Bildes nicht sauber abgetrennt. Es bedurfte daher einer Extraktion des Bildes aus dieser Datei, damit es seinem Anspruch gerecht ausgedruckt werden konnte.

Der Vollständigkeit halber sei schon hier gesagt, daß die Parameter Informationen über ein Halbtonbild (`-gy`), die Breite der Farbinformation (`-x4`), die Größe der Graupixel (`-u2`) und die Bildaufhellung (`-b2`) geben.

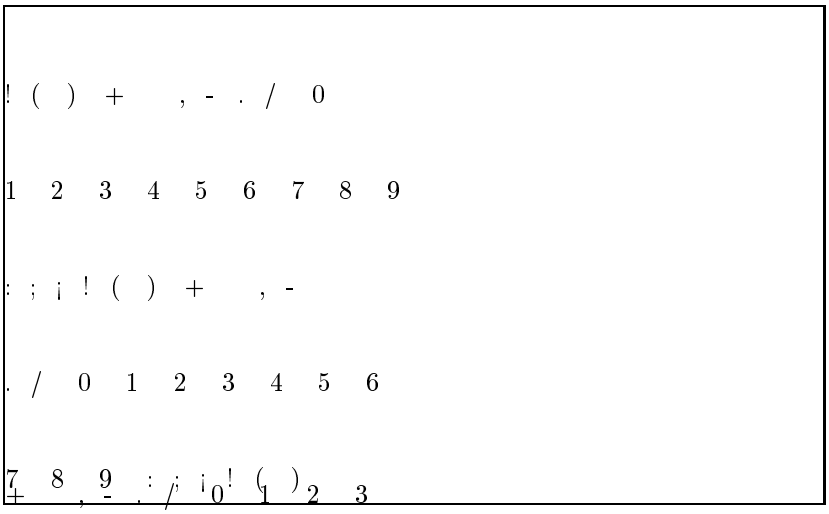


Abbildung 8: Beispiel zu einer Bitmap mit 16 Graustufen, ausgegeben mit 14 Graustufen

3 Das Verfahren der Integration

Wie bereits in der Einleitung erwähnt, wird das Bild von BM2FONT in eine für TEX und Treiberprogramme verständliche Form umgewandelt. Im einzelnen sieht die Lösung so aus, daß in einer Textdatei TEX die Dimensionen des Bildes bekanntgegeben werden und das Bild entsprechend den teilweise restriktiven Bedingungen von existierenden Treibern auf Zeichensätze verteilt wird.

Für das Beispiel in der Abbildung 2 erzeugt BM2FONT folgende Eingabedatei, die zuerst die neuen Fonts definiert, dann die Breite des Bildes feststellt,

anschließend das Bild in eine Box setzt und zum Schluß ein Makro definiert, das diese Box ausgibt.

```
\newbox\cobrabox
\newdimen\cobraw
\font\cobraa=cobraa
\font\cobrab=cobrab
\setbox\cobrabox=\vbox{\hbox{%
\cobraa !()+,-}}
\cobraw=\wd\cobrabox
\setbox\cobrabox=\hbox{\vbox{\hsize=\cobraw%
\parskip=0pt\offinterlineskip\parindent0pt%
\hbox{\cobraa !()+,-}%
\hbox{\cobraa ./0123}%
\hbox{\cobraa 456789}%
\hbox{\cobraa :;<\cobrab !()}}}
\ifx\parbox\undefined
\def\setcobra{\box\cobrabox}
\else
\def\setcobra{\parbox{\wd\cobrabox}{\box\cobrabox}}\fi
```

TEX erhält aus den TFM-Dateien die Informationen über die Dimensionen der Zeichen, die für die Bildkonvertierung benutzt wurden, die Treiber holen sich wie üblich die PK- oder Pixel-Dateien, um das Dokument samt Bild auszugeben.

Wie aus der Eingabedatei ersichtlich, werden die einzelnen Zeichen, die bis zu einem halben Zoll im Quadrat groß werden können, lückenlos aneinander gesetzt. Sofern also benutzte Treiberprogramme mit `maxdrift` arbeiten, um einen besseren Satz zu erreichen (was ohnehin nur bei niedrigen Auflösungen sinnvoll ist wegen der Auswirkungen von Rundungsfehlern), kann es zu nicht erwünschten überlappenden Stellen im Bild kommen. In diesem Fall sollte man einen anderen Treiber benutzen oder hoffen, daß `maxdrift` auf Null gesetzt werden kann, wie es in der neuen Treiberfamilie von Nelson Beebe der Fall sein wird.

Noch eine Bemerkung zu Treibern: BM2FONT stellt eine minimale Anforderung an Treiberprogramme, die sich auf die Größe das Zeichens

M

bezieht. Falls ein Treiber dieses Zeichen ausgeben kann, dann sollte er auch mit den Zeichensätzen, die BM2FONT erzeugt, keine Probleme haben, zumal obiges Zeichen sogar noch breiter ist als 1,27cm.

Üblicherweise sind in einer TEX-Installation verschiedene Unterverzeichnisse vorhanden, die meist durch Environmentvariable gesetzt sind. BM2FONT benutzt sie ebenfalls, um die erzeugten Dateien an die richtige Stelle zu schreiben. Die benutzten Variablen sind `texfonts, texinputs und dirpxl`. Hierzu ein Beispiel mit der Standardeinstellung für eine Auflösung von 300 dpi:

Env.-Variable	BM2FONT schreibt Datei
`texinputs=\tex\inputs`	`\tex\inputs\cobra.tex`
`texfonts=\tex\fonts\tfm`	`\tex\fonts\tfm\acobra.tfm`
	`\tex\fonts\tfm\bcobra.tfm`
`dirpxl=\tex\fonts\pxl`	`\tex\fonts\pxl300\acobra.pk`
	`\tex\fonts\pxl300\bcobra.pk`

Sind die Environment Variablen nicht gesetzt, so schreibt BM2FONT in das aktuelle Unterverzeichnis. Gleiches gilt, wenn die voreingestellten Unterverzeichnisse beim Eröffnen der Dateien nicht gefunden werden.

4 Variationen durch Parameter

Da nicht alle Bilder mit dem gleichen Verfahren erzeugt werden sollen, ist eine Anzahl von Parametern notwendig, die zum einen in Abhängigkeit von der Art des Bildes, der benutzen Ausgabegeräte und auch deren Auflösung gesetzt werden können.

Bei Aufruf des Programms ohne irgendwelche Angaben wird eine Auflistung der möglichen Parameter ausgegeben.

```
C:\>bm2font
This is BitMapTOfont, version 1.0 of july 91
Converting Bitmap Files to TeX-Fonts
usage is BM2FONT filename and parameters
-h<horizontal resolution>     (pixel/inch, std 300)
-v<vertical resolution>       (pixel/inch, std 300)
-l<length of mapline>         (in bytes, only pure bitmaps)
-a<show pictures on screen>   (y or n, std n)
-e<stretch EGA pictures>      (y or n, std y)
-i<inversion of pixels>       (y or n, std n)
-g<greypixels in bitmap>      (y or n, std n)
-p<write pixel files>         (y or n, std n)
-w<let white be light grey>   (y or n, std y)
-d<distribute errors>         (y or n, std y)
-s<separation of grey dots>   (y or n, std n)
-r<repeat each grey pixel>    (y or n, std n)
-u<pixels for grey rectangle> (less 8)
-c<vert. pixels for rectangle>(less 8)
-x<bits per sample>           (0 < x < 9)
-b<reduce halftone colors>    (f.e. by 1, less u*c*4, std 0)
-t<gradation value>           (in %, std 70)
-z<area of gradation>         (in %, std 70)

C:\>
```

An dieser Stelle muß gesagt sein, daß das Bild direkt für das letztendlich verwendete Ausgabegerät generiert werden soll. Die Geräteunabhängigkeit von TEX ist hier nicht mehr sinnvoll einzusetzen, da im Preview mit 102dpi Auflösung nur noch interessant sein kann, wo das Bild gesetzt wird. Eine Aussage, wie ein Halbtonbild auf einem Drucker mit 300dpi oder etwa auf einer Lichtsatzanlage mit 2400dpi aussehen wird, kann im Preview nicht gemacht werden. Es sind selbst auf den Druckern oft mehrere Tests notwendig, bis die Parameter optisch optimal gesetzt sind, wobei BM2FONT dazu eine kleine Hilfe anbietet.

4.1 Auflösung: Parameter `-h` und `-v`

Die Standardeinstellung für horizontale und vertikale Auflösung ist 300 Pixel pro Zoll. Mit `-h360` und `-v360` kann z.B. für einen gängigen 24-Nadeldrucker das Bild generiert werden. Da zur Berechnung der TFM-Informationen diese Angaben verwendet werden, kann hier nur die relative Größe des Bildes beeinflußt werden, die physische Größe wird von anderen Parametern beeinflußt, was allerdings nur in gewissen Maßen bei Halbtonbildern sinnvoll ist.

4.2 Verwendung von Bitmaps: Parameter `-l`, `-g` und `-x`

Diese Informationen sind nur bei reinen Bitmaps als Bildquelle anzugeben. Darauf ist bereits im Zusammenhang mit der Abbildung 8 eingegangen worden. Die Breite des Bildes wird mit `-l`xyz in Bytes angegeben. Sofern es sich um ein Halbtonbild handelt, in dem direkt die Grautöne enthalten sind, muß Parameter `-gy` gesetzt werden, da sonst von einer Schwarz/Weiß-Zeichnung ausgegangen wird. Bei Halbtonbildern ist auch die Bitbreite des Farbsample anzugeben, welches auf 8bit-Breite voreingestellt ist. Dabei werden dann 256 Farben bzw. Graustufen erwartet. Bei einem Bild mit 16 Farben liegt eine Breite von 4 Bits vor. In diesem Fall ist `-x4` anzugeben.

4.3 Grafikadapter: Parameter `-a`

Die Routinen zur Anzeige des Bildes und der einzelnen Zeichen der erzeugten Fonts sind so geschrieben, daß sie die gängigen Grafikadapterkarten unterstützen. Trotzdem kann es bei einigen VGA-Karten zu Problemen kommen. Aus diesem Grund wird standardmäßig die Videoausgabe unterlassen. Aktiviert wird die Anzeige mit `-ay`. In diesem Modus wartet BM2FONT auf die Betätigung einer Taste, nachdem das Bild angezeigt wurde. Die Taste `q` bewirkt den Abbruch des Programms, wobei die temporären Dateien erhalten bleiben.

4.4 Rasterweite: Parameter `-u` und `-c`

Zur Erzeugung von Halbtonbildern müssen die einzelnen Bildpunkte auf dem Papier so dargestellt werden, daß durch unterschiedlich große schwarze Punkte verschiedene Grautöne realisiert werden. Dazu werden für jeden Punkt aus der Bildquelle in einem Raster von bestimmter Größe entsprechend viele Pixel auf dem Papier geschwärzt. Dieses Verfahren wird Rastern oder Dithern genannt.

Die Qualität eines Halbtonbildes auf dem Papier ist unmittelbar von der Auflösung des Ausgabegerätes abhängig. Je höher die Auflösung ist, desto größer kann die Rasterweite gewählt werden, womit auch die Anzahl der darstellbaren Graustufen wächst. BM2FONT hat für die Auflösung des Ausgaberätes 300 Pixel/Zoll voreingestellt. Der Benutzer kann nun die horizontale (`-u`) und vertikale (`-c`) Rasterweite in Pixeln angeben. Diese angegebenen Pixel beziehen sich entgegen der üblichen Annahme jedoch auf ein Viertel des Rasters, in dem ein Bildpunkt gerastert wird. Wenn also zum Beispiel `-u3` angegeben wird (die vertikale Anzahl Pixel wird ebenfalls auf 3 gesetzt, solange mit `-c` kein anderer Wert angegeben wird), dann ergibt sich insgesamt ein 6x6-Raster für einen gedruckten Bildpunkt, der auf dem angenommenen Ausgabegerät eine Größe von 0.5mm hat.

Grundlage für das Rastern ist die folgende Dithermatrix. Die Zahlen in der Matrix stellen die Grauwerte dar, bis zu denen ein Pixel im Raster an den entsprechenden Positionen geschwärzt wird. Ausgegangen wird von den Grauwerten, die im Raster möglich sind.

	Spalte						
Reihe	1	2	3	4	5	6	7
1	01	03	06	10	18	26	38
2	02	04	07	12	19	28	40
3	05	08	09	14	21	30	42
4	11	13	15	16	23	32	44
5	17	20	22	24	25	34	46
6	27	29	31	33	35	36	48
7	37	39	41	43	45	47	49

Ein Rasterpunkt wird aus 4 benachbarten originären Pixeln gebildet. Dies bedeutet, daß jeder der Rasterweite entsprechende Grauwert in 4 Ausprägungen erscheinen kann. Die Anzahl der möglichen Grauwerte G ist also $G = 4uc$, wobei u der Rasterbreite und c der Rasterhöhe in Pixeln entsprechen. Bei einer Parametereinstellung wie zuvor erwähnt ergeben sich 36 Grauwerte mit je 4 Rastern, die von BM2FONT zur Erzeugung des Bildes benutzt werden.

Grauwert	Raster 1	2	3	4
1	●○○ ○○○ ○○○			
2	●○○ ○○○ ○○○	○○○ ○○○ ○○●		
3	●○○ ○○○ ○○○	○○○ ○○○ ○○●	○○○ ○○○ ●○○	
4	●○○ ○○○ ○○○	○○○ ○○○ ○○●	○○○ ○○○ ●○○	○○● ○○○ ○○○
5	●○○ ●○○ ○○○	○○○ ○○○ ○○●	○○○ ○○○ ●○○	○○● ○○○ ○○○
6	●○○ ●○○ ○○○	○○○ ○○● ○○●	○○○ ○○○ ●○○	○○● ○○○ ○○○
⋮				

Grauwert	Raster 1	2	3	4
20	●●○ ●●○ ●○○	○○● ○●● ○●●	●○○ ●●○ ●●○	○●● ○●● ○○●
21	●●● ●●○ ●○○	○○● ○●● ○●●	●○○ ●●○ ●●○	○●● ○●● ○○●
22	●●● ●●○ ●○○	○○● ○●● ●●●	●○○ ●●○ ●●○	○●● ○●● ○○●
⋮	⋮	⋮	⋮	⋮
34	●●● ●●● ●●●	●●● ●●● ●●●	●●○ ●●● ●●●	●●● ●●● ○●●
35	●●● ●●● ●●●	●●● ●●● ●●●	●●● ●●● ●●●	●●● ●●● ○●●
36	●●● ●●● ●●●	●●● ●●● ●●●	●●● ●●● ●●●	●●● ●●● ●●●

Pro Bildzeile werden die Raster r_i nach folgendem Schema verwendet:

r_1	r_4	r_1	r_4	r_1	r_4	...
r_2	r_3	r_2	r_3	r_2	r_3	...
r_4	r_1	r_4	r_1	r_4	r_1	...
r_3	r_2	r_3	r_2	r_3	r_2	...
r_1	r_4	r_1	r_4	r_1	r_4	...
r_1	r_4	r_1	r_4	r_1	r_4	...

4.5 Gradation: Parameter `-t` und `-z`

Da nahezu alle Ausgabegeräte dazu neigen, im dunkleren Bereich der Grauwertskala zu schnell zu schwarz zu werden (Tonwertzuwachs), ist es erforderlich, dieses Verhalten auszugleichen. Dies geschieht mit Hilfe der Gradation, der Korrektur des Tonwertzuwachses.

Die folgende Funktion beschreibt den Verlauf der Ausgabewerte der Halbtonpunkte, die in Abhängigkeit von den Parametern `t` und `z` variiert wird. Im dunklen Bereich ($x = 0$ entspricht Schwarz) erfolgt eine starke Aufhellung, die im helleren Bereich solange abnimmt, bis sie am Punkt x_0 verschwindet.

$$f(x) = \begin{cases} \frac{1}{2x_0^\alpha} x^{1+\alpha} + \frac{1}{2} x_0^\alpha x^{1-\alpha} & 0 \le x < x_0 \\ x & x \ge x_0 \end{cases}$$

$$\text{mit } \alpha = \tfrac{t}{100},\ x_0 = \tfrac{z}{100}$$

Mit der Standardeinstellung von 70 für `t` und `z` erzeugt BM2FONT ein Bild mit einer schwachen Aufhellung, die sich auf 70% des Grauwertbereichs auswirkt.

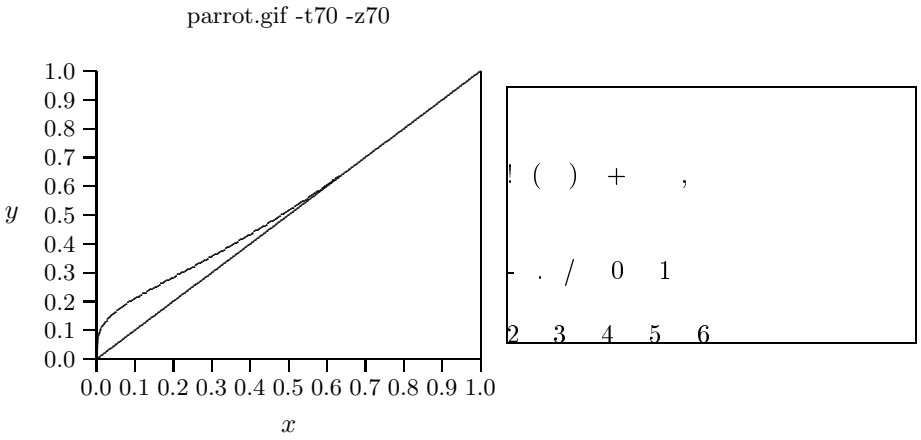


Abbildung 9: Beispiel zur Gradation mit Standardwerten, Ausgabe des Bildes mit 16 Graustufen

Mit einer stärkeren Aufhellung, die auf 90% der Grauwertskala wirken soll, ergibt sich ein deutlich sichtbarer Unterschied nicht nur im Kurvenverlauf sondern auch bei dem gedruckten Bild.

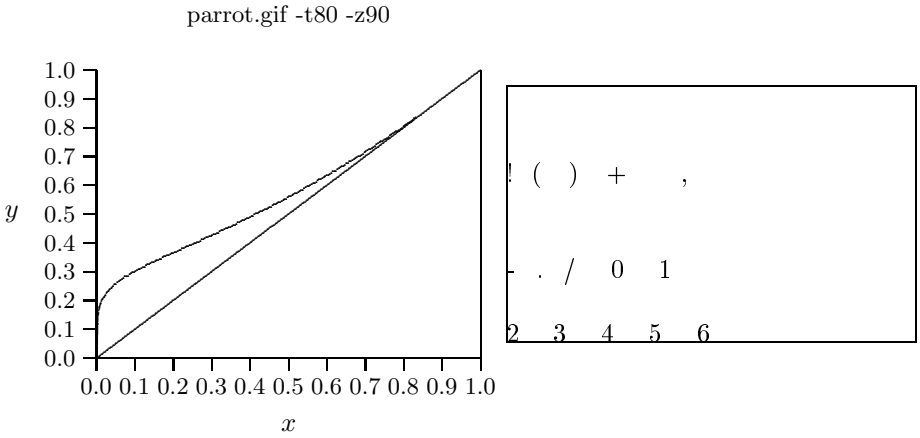


Abbildung 10: Beispiel zur Gradation mit Änderung von Aufhellungsfaktor und Wirkungsbereich, Ausgabe des Bildes mit 16 Graustufen

Die Korrektur des Tonwertzuwachses wird nicht durchgeführt, wenn BM2FONT mit `-t0` aufgerufen wird. Dies ist unter Umständen dann angebracht, wenn eine Aufhellung mit anderen Mitteln erreicht werden soll. Auch bei einer geringen Anzahl von Graustufen (8 oder 16) in der Bildquelle ist die Gradation nicht sehr sinnvoll.

4.6 Globale Aufhellung: Parameter `-b` und `-s`

Falls ein Bild insgesamt zu dunkel erscheint, kann eine Aufhellung dadurch erreicht werden, daß die originären Farbwerte nicht auf die gesamte Breite der Ausgabeskala verteilt werden oder zusätzliche weiße Pixel die Rasterpunkte trennen.

Mit dem Parameter `-b` wird die Anzahl der zu benutzenden Graustufen bei der Ausgabe um die angegebene Zahl reduziert. Bei einem 6x6-Raster und

möglichen 36 Graustufen erzeugt BM2FONT mit der Einstellung `-b3` ein Bild mit 33 Graustufen. Dieser Parameter stellt dann einen Notbehelf dar, wenn das Bild tatsächlich zu dunkel ist, da diese Maßnahme die Bildinformation weiter reduziert. Besser wäre in diesem Fall eine entsprechende Aufbereitung des Bildes bei der optischen Bilderfassung durch Veränderung des Kontrastes oder der Helligkeit.

In der Regel ist es aber oft so, daß bei den erfassten Bildern nicht das gesamte Farbintervall genutzt wird. BM2FONT versucht jedoch bei der Bilderzeugung die vollständige Skala an Graustufen zu nutzen, was zu übermäßig starken Kontrasten führen kann. In diesem Fall kann die Aufhellung mit `-b` eine bessere Bildqualität liefern. Das Programm gibt bei diesem Sachverhalt einen Hinweis auf die mögliche Verbesserung durch Aufhellung des Bildes.

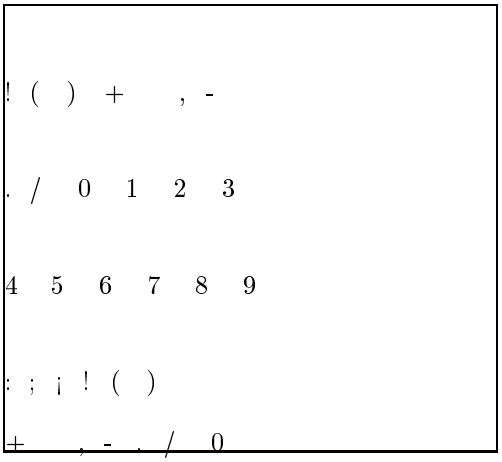
Es ist allerdings auch möglich, daß ein Laserdrucker die Tonerpartikel nicht so klein, wie es der Auflösung entsprechen würde, auf das Papier aufträgt. Diese Unzulänglichkeit kann durch die Angabe `-sy` ausgeglichen werden. Das geschieht durch die Separation der erzeugten Rasterpunkte mittels weißer Pixelspalten und -zeilen.

4.7 Fehlerverteilung: Parameter `-d`

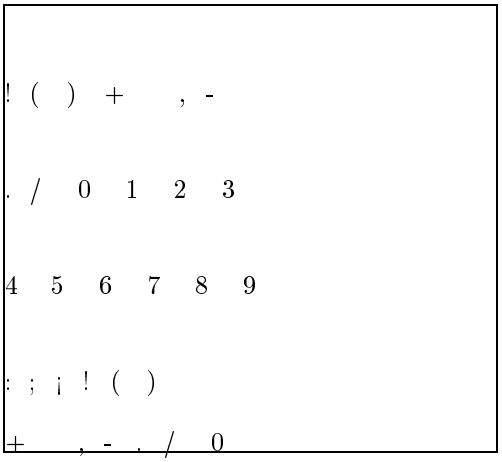
Bei der Erzeugung von Halbtonbildern ist man in der Regel mit dem Problem konfrontiert, daß die Anzahl der Graustufen in einem Bild von der Anzahl der Graustufen des Pixelrasters abweicht. Rein mathematisch wird das durch die Rundungsfehler bei der Umrechnung der Farbwerte deutlich. Ein bekannter Algorithmus (Floyd/Steinberg) löst dieses Problem, indem die Rundungsfehler kumulativ auf die benachbarten Bildpunkte verteilt werden. Durch das besondere Vorgehen bei der Rastergenerierung von BM2FONT ist dieses Verfahren in Annäherung bereits realisiert.

Trotzdem ist in manchen Bildern noch der Informationsverlust durch Rundungsfehler sichtbar, die als starke Farbsprünge sichtbar sind. Zur Glättung berücksichtigt BM2FONT nach der Umrechnung zeilenweise den gemittelten Wert – einschließlich seines Rundungsfehlers – benachbarter Bildpunkte mit

unterschiedlichen Farbwerten. Dadurch kann in den meisten Fällen eine deutliche Verbesserung der Bildqualität erzielt werden. Dazu ein Beispiel, bei dem circa 4% der Grauwerte durch die Fehlerverteilung verändert wurden:



Bildgenerierung mit Fehlerverteilung



Bildgenerierung ohne Fehlerverteilung

Standardmäßig ist die Durchführung der Fehlerverteilung vorgesehen. Mit `-dn` kann das Verfahren unterdrückt werden. Dies ist zum Beispiel bei solchen Bildern wie dem auf der Titelseite anzuraten.

4.8 PK–Format oder Pixel: Parameter `-p`

Mit der Einstellung `-py` schreibt BM2FONT statt gepackter Fonts das inzwischen veraltete PXL-Format. Da unter Umständen auf anderen Systemen Treiber existieren können, die das PK-Format nicht unterstützen, kann so eine notwendige Konvertierung eingespart werden.

An dieser Stelle sei nochmals darauf hingewiesen, daß BM2FONT die einzelnen Zeichensätze eines Bildes bis zu einer Größe von 64KB generiert. Diese Einschränkung bezieht sich auf die Größe eines Fonts im PXL-Format, nicht auf das PK-Format, weil nicht auszuschließen ist, daß einige Treiber aus leicht nachvollziehbaren Gründen eine Konvertierung vom PK- auf das PXL-Format vornehmen könnten. In solchen Fällen treten dann keine Probleme bei dynamischer Speicheranforderung für Fonts auf.

4.9 Anpassung an X/Y–Verhältnis: Parameter `-e`

Die Vielfalt an Grafikkarten mit unterschiedlichen Bildschirmauflösungen hat dazu geführt, daß auch in vielen Bitmap-Formaten eine Information über das Verhältnis zwischen Breite und Höhe eines Pixels enthalten ist. Dadurch können beim Austausch von Bildern zwischen Systemen mit unterschiedlichen Grafikkarten Verzerrungen in horizontaler oder vertikaler Richtung ausgeglichen werden.

Wird zum Beispiel ein Bild, das auf einem PC mit einer EGA-Karte (640x350 Pixel) erzeugt wurde, auf einem anderen System mit einer VGA-Karte (640x480 Pixel) ohne Ausgleich in der Vertikalen dargestellt, so füllt es nur einen Teil des Bildschirms aus. Die Reaktion der betrachtenden Person ist dann nicht vorhersehbar.

Dieses Problem ist auch bei der Darstellung eines Bildes auf Papier vorhanden. BM2FONT führt anhand der Informationen über die X/Y-Relation der Pixel eine Anpassung durch. Falls dadurch eine wesentliche Veränderung der Bildqualität auftritt, kann diese Anpassung durch `-en` unterdrückt werden. In diesem Fall kann mit entsprechender Einstellung von `-u` und `-c` manuell eine Anpassung vorgenommen werden.

4.10 Sonstiges: Parameter `-i`, `-w` und `-r`

Üblicherweise wird die Farbe Schwarz in den bei Bildformaten verwendeten Paletten mit dem Wert 0 dargestellt. Sollte es einmal nicht so sein, dann kann mit `-iy` ein Negativ erzeugt werden. Die Farbpalette wird also invertiert.

Bei Halbtonbildern wird ein besserer Gesamteindruck erzielt, wenn auch die weißen Flächen auf dem Papier hellgrau sind. Falls diese leichte Farbveränderung nicht gewünscht ist, dann läßt `-wn` weiße Flächen tatsächlich weiß aussehen.

Im Abschnitt über Rasterweiten wurde bereits erwähnt, daß aus vier benachbarten Bildpunkten ein Rasterpunkt gebildet wird. Bei relativ kleinen Bildern und einem hochauflösenden Ausgabegerät kann die generierte Abbildung leicht zu klein geraten. In diesem Fall vergrößert die Parametereinstellung `-ry` die Ausgabe ohne Veränderung der Rasterweiten durch Wiederholung der einzelnen originären Bildpunkte.

5 Ein Beispiel zur Anwendung

Die Beschreibung der Parameter von BM2FONT bedürfen sicherlich noch einer Ergänzung hinsichtlich ihrer Auswirkungen. Dies trifft insbesonders dann zu, wenn verschiedene Parameter mit unterschiedlichen Angaben kombiniert werden. Da es sich um die Erzeugung von Bildern handelt, lassen sich die Auswirkungen durch die Bilder selbst zeigen.

Als Grundlage für die Variationen dient das Foto eines sicherlich sehr bekannten Gebäudes:

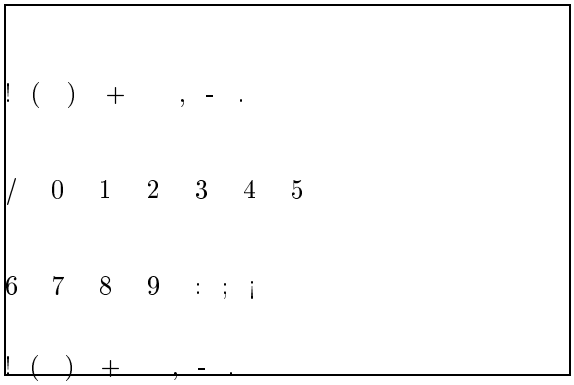


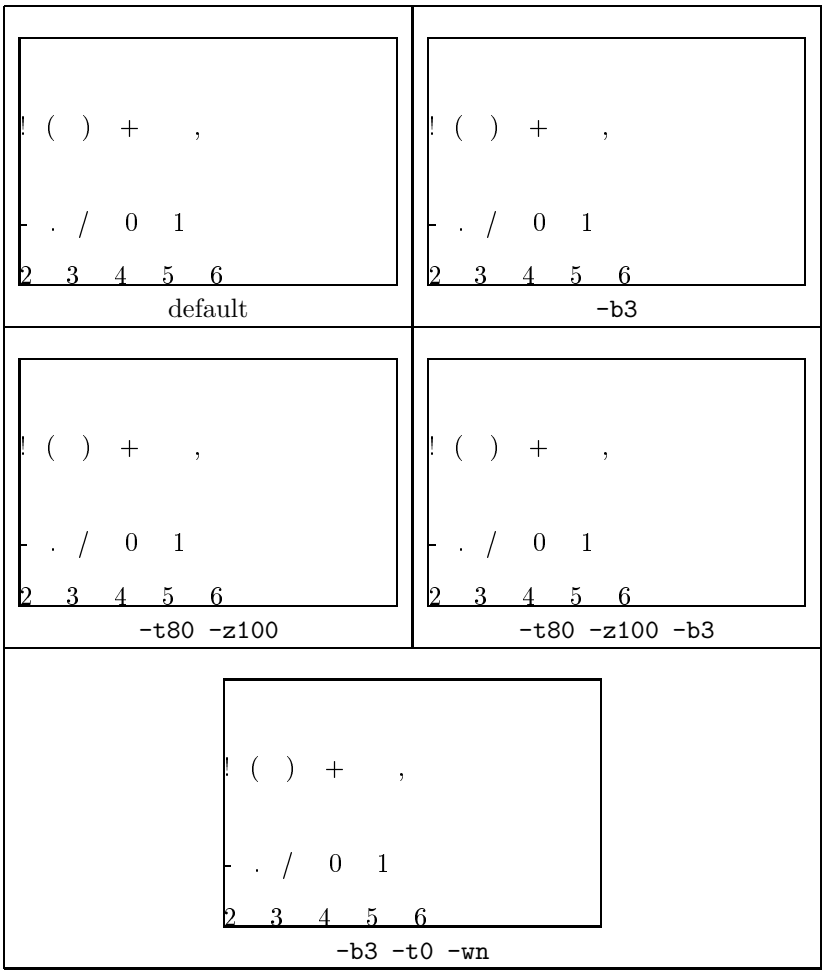
Abbildung 11: Bild des Capitols, ausgegeben mit 36 Graustufen, Gradation verstärkt mit Wirkung auf ganzen Grauwertbereich

Es wurde nicht nur deswegen gewählt, weil es symbolisch für das Land steht, in dem TEX das Licht der Welt erblickte. Die zugrunde liegende Bildquelle wirft gewisse Probleme bei der Umwandlung in ein Halbtonbild auf, die in den einzelnen Variationen zu offensichtlichen Unterschieden beitragen. Eine Lupe ist nicht erforderlich.

Die folgenden Bilder des Capitols wurden alle mit 16 Graustufen gedruckt. Unter den Abbildungen sind die jeweils verwendeten Parametereinstellungen angegeben. Bis auf das Bild mit der Parametereinstellung `-t0` ist bei allen Bildern die Korrektur des Tonwertzuwachses durchgeführt worden. Auf die Darstellung der Auswirkungen der Fehlerkorrektur wurde bewußt verzichtet, da dazu nochmals mindestens 5 Beispiele erforderlich gewesen wären.

BM2FONT wurde wie folgt aufgerufen, wobei die Variationen der Parameter unter den jeweiligen Abbildungen zu finden sind:

```
bm2font capitol.295 -l295 -gy -x8
```



6 Hardware und Software

BM2FONT läuft auf allen dem Industriestandard kompatiblen PC's unter dem Betriebssystem MS-DOS. Die erzeugten Ausgaben sind mit allen TEX-

Versionen verarbeitbar.

Es werden VGA-, EGA- und Herkules-Grafikkarten unterstützt. Bei Problemen mit Grafikadaptern ist die Parametereinstellung `-ay` nicht zu verwenden.

Kontaktadresse bei andauernden Problemen:

Heinrich-Heine-Universität Düsseldorf
Universitätsrechenzentrum
Friedhelm Sowa
Universitätsstraße 1
4000 Düsseldorf 1
Telefon: 0211 / 311-3913
Telefax: 0211 / 311-2539
Email: tex@dd0rud81.bitnet oder sowa@convex.rz.uni-duesseldorf.de