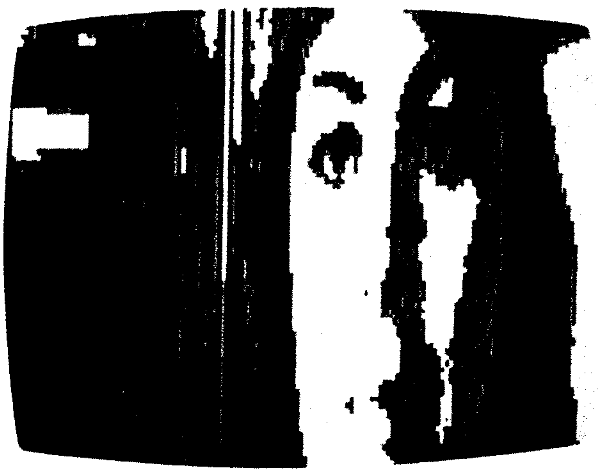
BY LLOYD BURCHILL

VIDEO STRETCH

Rubber visuals in ACTION!



Stretch visual images like silly putty on any 8-bit Atari with at least 48K memory and a disk drive. Video Stretch requires the ACTION! language cartridge from Optimized Systems Software. (Disk subscribers this month will find a runtime version that doesn't require ACTION!) Paddle controllers are recommended.

Outstanding new graphics possibilies are still being discovered on the Atari 8-Bit computers. The Video Stretch program will vertically compress or stretch an entire screen and select different parts of it to view—in real time. It's a spectacular, eye-catching effect.

Video Stretch can be used for an impressive slide-show program, or just to see how your favorite screens look when warped all out of shape. In any case, the program demonstrates that the 8-bit Atari is still capable of surprising new feats.

The program will work with its own Graphics 9 sample picture. Or you can use your own pictures made with Micro-Painter, Graphic Master and Computereyes. If you have none of these, use *Rapid Graphics Converter* (**Antic**, November 1985) to change your images into a compatible format.

TRYING IT OUT

You will need a pair of paddle controllers plugged into port 1. Paddle 0 controls the size of the display and paddle 1 selects the portion of the image to be seen. You can also use this program with a touch tablet, but it doesn't work quite as well as the paddles. And of course you can write your own routines to send values to Stretch( ).

Antic Disk subscribers: This month as a bonus you'll find a "runtime" version of Video Stretch that operates without the ACTION! cartridge. Follow the disk Help file instructions for loading STRETCH.EXE. This listing was too long to print in the magazine, and it cannot be adapted by the user as the ACTION! source code can be.

Carefully type in Listing 1, STRETCH.ACT, following the instructions in the ACTION! manual and

continued on page 44

VIDEO STRETCH
continued from page 37

save a copy before you run it. If you just want to see Video Stretch operate on the Graphics 9 demo picture, type D from the main menu.

If you have some 62-sector microscreens to try it with, choose the L option from the main menu. The program will display any 62-sector file named D:PICTURE. You can change this default name by altering the filename in the Load( ) procedure.

You may alter the program to accept any 192-line graphics mode. To make the program accept pictures created in Graphics 15, for example, just redefine IR at the beginning of the program as 14, and change the graphics call in Rubber_Band to Graphics(15). If you have a non-XL computer, Graphics(15) is not available. You can replace it with Graphics(8), and the display will fix itself as soon as Stretch( ) operates.

Similar changes can be made to the program to make it work on any 192-line graphics mode, namely Graphics 8, 9, 10, 11, 14, and 15. For the little-used Graphics 14, change a*40 in Sinit( ) to a*20.

HOW IT WORKS

The program is written in ACTION! from Optimized Systems Software, which is ideal for the combination of speed and arithmetic that is required. The Stretch( ) routine needs two parameters; the first is the size, in scan lines, of the image that will appear on the screen. Normal size is 192 lines. You can choose any size from one to about 500 lines and the image will be proportioned correctly.

Since an image bigger than 192 lines can't be displayed all at once, the second parameter tells what line of the original image will appear at the top of the screen. (It should be between 0 and 191.) This allows you to scan any portion of the expanded image. It also works with images that are smaller than full-size, allowing you the interesting capability of expanding the picture and moving it up and down.

DETAILS

The Stretch( ) procedure uses several line-drawing algorithms and speedy integer arithmetic to move each scan line to its proper place on your screen.

Before entering Stretch( ) you must initialize it with a call to Sinit( ). A space of 580 bytes is required for the complex display list generated by the program. Each line has an independent LMS instruction.

Because the new display list is so long, your choice of places to put it is limited to the first 443 bytes in any given kilobyte. If you are using high-resolution Player/Missile graphics, those first three unused pages of P/M space are ideal.

Lloyd Burchill is a high school senior from Newcastle, New Brunswick who likes to write programs with an artistic flair. In 1985 he won a national award for a program about the moons of Uranus.

HARD DISKS
continued from page 36

floppy disk will present one picture about every four seconds, for an average speed of (32K/4) 8K bytes per second.

Our results here had me scratching my head. SHOPIC displayed pictures at slightly faster than two frames per second (fps) on the empty SupraDrive, the SupraDrive with files and the Atari Hard Disk. This means they're moving nearly 64K bytes per second (2 * 32K). But our HabaDisk ran at half that speed—a little less than one fps—and we're still not sure why.

HARD DISK UTILITIES

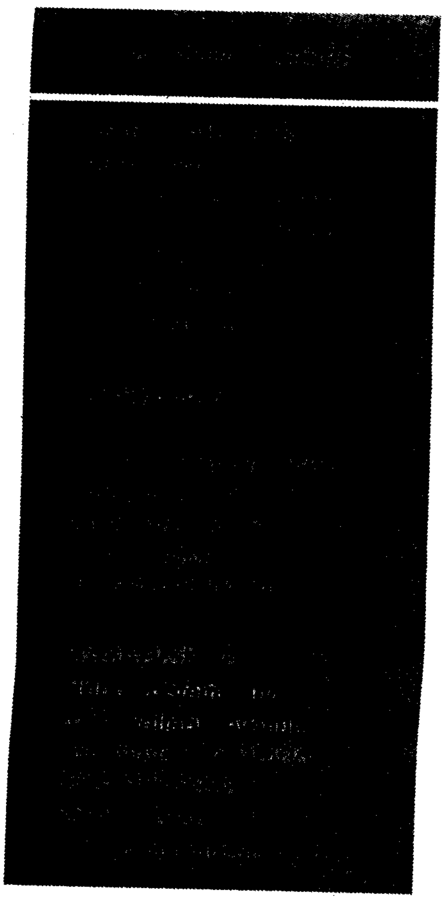
Every ST hard disk is supplied with at least four utility programs. Each manufacturer has slightly different programs, but they all do much the same job:

1. Format the hard disk. This program acts much like the format disk program for the floppy disks. Like the floppies, a hard disk is born blank, and needs to be told how and where to store information on itself. You may divide one "physical" hard disk into as many as four "logical" hard disks, each with its own drive letter identifier—C, D, E, etc.

2. Boot the hard disk. This program, when placed inside an AUTO folder, teaches the Atari ST how to communicate with the hard disk when powered up. It may also be used outside the AUTO folder, after the Atari ST is up and running.

3. Clear directories. Formatting a hard disk can take as long as 10 minutes. A faster way to "erase" a hard disk is just to re-format the disk directories. In effect, the disk "forgets" where everything is stored on the disk and pretends the disk is blank, but formatted. This operation takes only seconds to perform.

4. Park the Read/Write head. This means running a program which locks the R/W head in a safe position (usually above the innermost track) where it won't be jostled against data track surfaces if the hard disk is
continued on page 52


[illegible]

eye-catching horizontal display

TEXT SIDEWINDER

Article on page 19

LISTING 1

Don't type the TYPO II Codes!

```
SI 2 REM TEXT SIDEWINDER
ST 3 REM BY JESS ENGELHART
FS 4 REM (c) 1986, ANTIC PUBLISHING
NF 5 GRAPHICS 0:? "MODE (0, 1 OR 2)";:INP
   UT MODE:IF  NOT (MODE=0 OR MODE=1 OR M
   ODE=2) THEN 5
QM 10 GOSUB 600:GOTO 1000
XB 50 IF PEEK(764)<>33 THEN 60
YQ 55 TIME=TIME-10:IF TIME<10 THEN TIME=1
   50
JZ 60 ? TIME:POKE 764,255:FOR R=0 TO TIME
   :NEXT R:RETURN
XK 600 GRAPHICS MODE:POKE 752,1:? "▚":POS
   ITION 1,1:? #6;"PRESS SPACE BAR TO  AL
   TER BANNER speed"
LE 610 POKE 703,4:REM CREATE A TEXT WINDO
   W IN GR.0
OL 620 DIM A$(2500),UU$(120),VV$(120),WW$
   (120),YY$(120),ZZ$(120):RESTORE 4000:R
   EAD A$,UU$,VV$,WW$,YY$,ZZ$
QU 625 A$(LEN(A$)+1)=UU$:A$(LEN(A$)+1)=VV
   $:A$(LEN(A$)+1)=WW$:A$(LEN(A$)+1)=YY$:
   A$(LEN(A$)+1)=ZZ$
XU 630 TIME=150:LL=19:POSITION 3,8:? #6;"
   ANTIC BACK then"
GD 640 IF PEEK(87)=0 THEN LL=38:REM IF GR
   .0 THEN LL=38
RG 645 GOSUB 50:RETURN
```

```
TG 1000 REM BANNER PRINT ROUTINE 1000 TO
   1500
JT 1001 FOR P=0 TO LL:POSITION LL-P,8:? #
   6;A$(1,P+1):GOSUB 50:NEXT P:POKE 77,0
HU 1010 FOR P=P+1 TO (LEN(A$)):POSITION 0
   ,8:? #6;A$(P-LL,P):GOSUB 50:NEXT P
IW 1012 A$(LEN(A$)+1)=CHR$(32)
JM 1015 FOR Q=P-(LL+1) TO P:POSITION 0,8:
   ? #6;A$(Q,P):GOSUB 50:NEXT Q
MU 1020 GOTO 1000
TB 4000 DATA FOR ANTIC IT ALL STARTED ON
   A kitchen table IN jim capparell's POT
   RERO HILL APARTMENT IN SAN FRANCI
ZC 4002 DATA SCO. "WHEN JIM SAID 'HEY! LE
   T'S START A MAGAZINE!' I SAID 'SURE. W
   HY NOT?' RECALLS marni tapsco
KM 4004 DATA tt (antic's CO-FOUNDER AND A
   RT DIRECTOR. "I WOULD NEVER HAVE IMAGI
   NED THEN THAT TODAY -- FOUR YEA
TG 4006 DATA RS LATER -- I'D BE MAKING A
   REAL salary."  "marni SUGGESTED A COMP
   UTER NAME SUCH AS byte." SAY
AE 4008 DATA S capparell.  "SO THAT GOT M
   E THINKING ABOUT COMPUTER STUFF AND OU
   T POPPED ANTIC -- SHORT FOR al
AG 4010 DATA pha numeric telivision integ
   rated circuit -- ONE OF THE 8-BIT ATAR
   I SPECIAL CHIPS.
```

rubber visuals in action!

VIDEO STRETCH

Article on page 37

LISTING 1

```
;STRETCH
;BY LLOYD BURCHILL
;(c) 1986, ANTIC PUBLISHING

Module

byte key=764,IR
card scr=88,dlist=560

define dlspace="14592"
define dlsend= "14592 +575"
 ;a 580 byte long space is needed
 ;that includes no addresses on
 ;a 1K boundary
 ;(excepting first byte)

card array memline(192)

Proc Stretch(card lines,card vstart)

;Parameters:
;'lines' is number of scan lines the image sh
ould occupy on the screen
;'vstart' is scan line of original image that
 will appear at top of screen
```

```
Byte pointer p
card pointer q
card bigline,addr,inc,temp

inc=24576/lines

bigline=(inc rsh 1)+(vstart lsh 7)

p=dlspace+3
q=dlspace+4

do

 temp=bigline rsh 7
 addr=memline(temp)
 p^=IR+64
 p==+3

 q^=addr
 q==+3

 bigline==+inc
 if bigline>24576 or p>dlsend then exit fi

od
```

```
p^=65

Return


Proc Sinit()        ;initialize
card a

setblock(dlspace,3,112)

for a=0 to 191 do
memline(a)=scr+a*40 od
;change to 'a*20' to use mode 14

stretch(192,0)
dlist =dlspace

Return


Proc Load()         ;load disk picture
byte pointer p

Close(5)
Open(5,"D1:PICTURE",4,0)

for p=scr to  scr+7679
do
 p^=GetD(5)
od

 p=712
 p^=GetD(5)

for p=708 to  710
do
 p^=GetD(5)
od

Close(5)

Return


Proc Drawing()      ;example picture
byte t,u,v,w,x,y,z

for w=0 to 30
do
 color=rand(16)
 x=rand(76)
 y=rand(180)
 for z=0 to 3
  do
  Plot(x+z,y) Drawto(x+z,y+12)
  od
od

for x=0 to 14
do
 color=x+1
 Plot(0,x) Drawto(79-x,95) Drawto(0,191-x)
 Plot(0,29-x) Drawto(52+x,95) Drawto(0,163+x)

od

for w=0 to 15
do
  z=10+ rand(70)
  y=rand(192-z)
  u=rand(20)
  v=rand(50-u)+15
  t=10+rand(21)

  for x=0 to 14
  do
   color=(15-x)*t/30
   Plot(v-x,y+z)
   Drawto(v-10+u,y)
   Drawto(v+x,y+z)
  od
 od

Return


Proc Rubber_Band() ;main procedure
byte pad1=624,pad2=625,mode
card h

do
   IR=15
   ;use 15 for modes 8,9,10,11
   ;use 14 for mode 15, "graphics 7 1/2"
   ;use 12 for mode 14  (160 x 192 x 2 colors
)

  Graphics(0)
  Poke (752,1)
  PrintE ("Load your own picture")
  PrintE ("Demo mode")

  do
     until key=0 OR key=58
  od
  mode=key
  Poke(764,255)

   if mode=58 then
     Graphics(9) Poke(712,128)
     Drawing()

   elseif mode=0 then
     Graphics(8) Poke(712,128)
     IR=14
     Load()
   fi

  ;Use either Load() or Drawing()
  ;and adjust graphics call
  ;according to preference

  Sinit()
  do

   h=pad1

   if pad2<192 then
    Stretch(h lsh 1 ,pad2)
   fi

  until key<>255
  od

until 0=1
od

Return
```

game of the month

SSSNAKE!!!

Article on page 23

LISTING 1

Don't type the TYPO II Codes!

```
PW 1 REM SSSNAKE
EU 2 REM BY CHET WALTERS
FR 3 REM (c) 1986, ANTIC PUBLISHING
IV 10 GOTO 4000
UY 20 IF PEEK(53279)=N6 THEN 2000
HO 25 S=STICK(0):IF PEEK(KEY)<>TFF THEN P
   OKE KEY,TFF:GOSUB 700
TV 30 X=X+STX(S):IF X>ETN THEN X=ETN:GOSU
   B 500
ZN 35 IF X<N THEN X=N:GOSUB 500
NF 40 Y=Y+STY(S):IF Y>20 THEN Y=20:GOSUB
   500
AL 45 IF Y<N THEN Y=N:GOSUB 500
```

continued on next page