Manual for ScopeShapes 0.1

Chris Meighan

Manual for ScopeShapes 0.1

Chris Meighan

Published 2005

Table of Contents

Introduction

What is ScopeShapes?

ScopeShapes is a piece of computer software which uses the stereo sound output present on most modern computers to drive the horizontal and vertical axes of any oscilloscope which has an X - Y mode, in order to produce 2D and 3D vector graphic images.

It reads DXF (Drawing eXchange Format) files, as produced by AutoCAD and many other programs, and can switch between up to 10 shapes. It can automatically rotate shapes in 3D space, and morph between different shapes.

How does ScopeShapes work?

Most desktop and laptop computers possess a stereo sound output, usually in the form of a 3.5mm jack socket. It can usually output at a resolution of 16 bits, at up to 44.1KHz. What this is, in effect, is a dual digital to analogue converter, optimised for the audio frequency spectrum.

An oscilloscope in X - Y mode (a function usually used to measure phase difference between two different signals) can be used a vector (as opposed to raster) graphic display by driving the two inputs with analogue voltages of a suitable magnitude. A varying voltage on the X input will drive the beam left or right (according to its polarity), whilst a voltage on the other will drive it up or down.

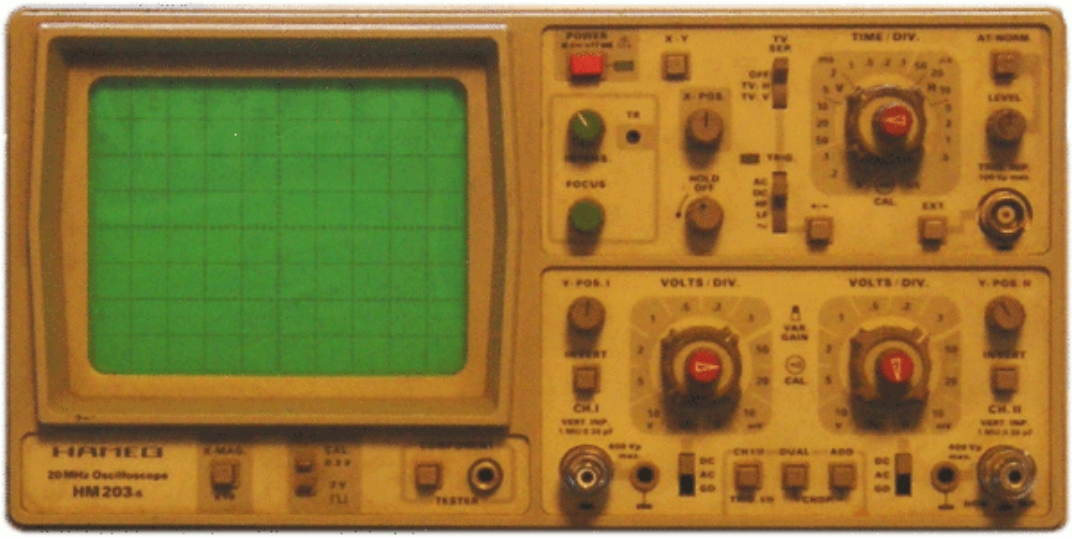
If we then connect the left and right side of the computer's stereo output to the X and Y inputs respectively (or the other way round, it doesn't really matter), and write some software to drive the outputs according to the geometry of a series of lines, we can draw shapes on the oscilloscope screen. That is the purpose of ScopeShapes.

Chapter 1. Using ScopeShapes

Connecting up the oscilloscope

As indicated in the introduction, the oscilloscope needs one channel to be connected to the left output of the computer's audio connector, and one to the right.

Exactly how the inputs are labelled will depend on your particular model, but the picture below shows the arrangement for mine, a Hameg HM203-6. Most models are unlikely to vary much from this.



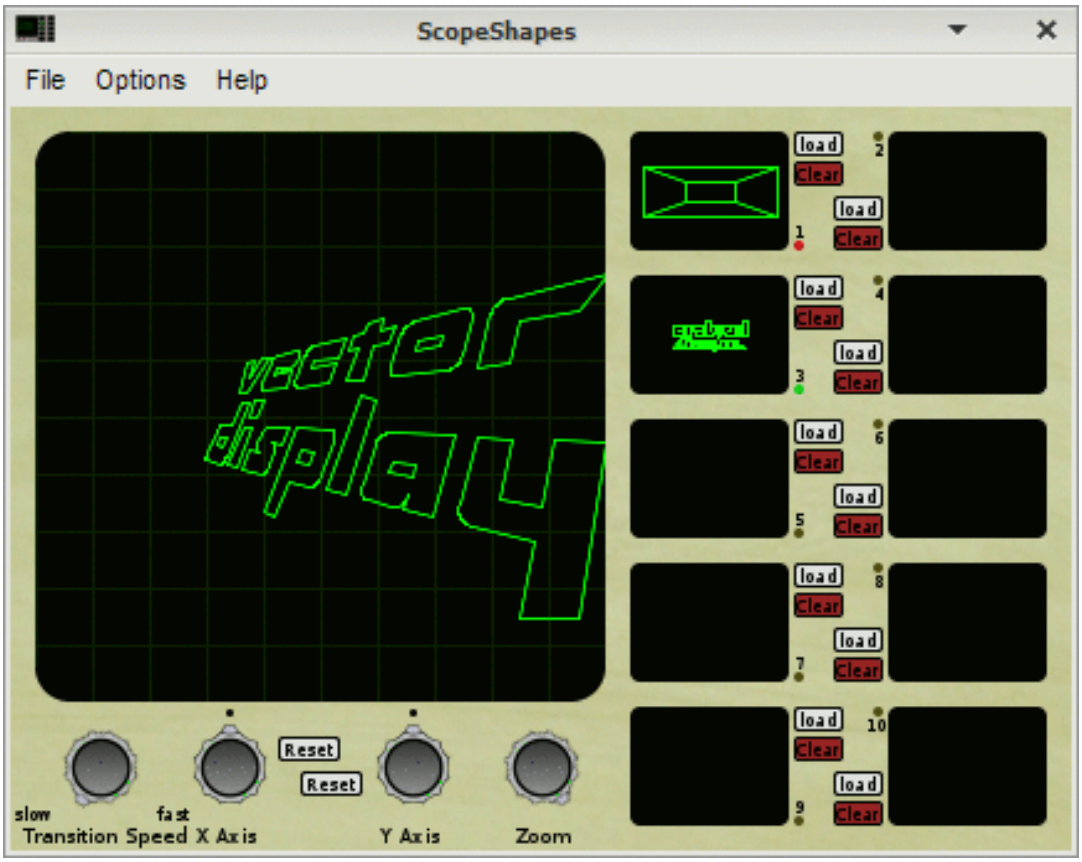
Since most scopes use BNC style connectors, the most convenient way to link it up to the computer (without the use of a soldering iron) is to use a suitably long 3.5mm jack to 2 x phono cable, and 2 BNC to phono adapters. Such things should be readily available in your local Maplin (UK), Radio Shack (US), or local equivalent.

Last but certainly not least, the 'scope should be switched into X - Y mode. This will usually be by pressing a button as on the machine shown above, or by a special setting on the time base control.

Once the software is running, it may be necessary to adjust the Volts/Div controls (usually both channels will be set to the same range). Mine seems to work best at somewhere between 50 and 200mV (millivolts) per division.

The main window

The main ScopeShapes window looks like this (the exact appearance will depend on the platform it is running on).



Before any shapes appear on the oscilloscope screen, it is likely that the sound card's parameters will need to be set up - see Hardware options below.

The large screen shows the shape that is currently (hopefully) showing on the 'scope screen. The ten smaller screens on the right are 'banks', which hold shapes which can be switched into view by clicking on them.

To load a shape into a bank, either click the white 'load' button next to it, or drag a file onto its screen. Any valid ASCII .dxf file will do - don't worry about scale and positioning, it will be automatically normalised to fit the screen. To clear a bank, click on the red 'clear' button below the 'load' button.

Note

You can specify a list of files on the command line when loading ScopeShapes, or drag some files onto the program's icon (which achieves the same thing, anyhow) to have them loaded on startup.

The row of dials underneath the main screen are used to alter various parameters;-

Table 1.1. Main window controls

Transition Speed	This affects how quickly one shape 'morphs' into another when a new bank is selected.
X axis and Y axis	These control how fast and in which direction the shape will be rotated about the 3D X and Y axes.
Zoom	This moves the viewpoint away or toward the shape.

Further control of the display can be effected via selecting 3D options from the Options menu.

Importing DXF files

Note

ScopeShapes will only read ASCII (NOT BINARY) DXF files.

First things first - simple is best.

The fundamental weakness in trying to use a soundcard's outputs to generate a graphical display is that it has insufficient bandwidth to do so very effectively. In simpler terms, the signal just can't change at a fast enough rate for complex shapes to be plotted in a reasonably short space of time. What this means is that overly-complex shapes will appear flickery and indistinct.

There is a way around this (sort of) - the Hardware options dialogue allows the speed of the beam to be adjusted, which is something of a trade-off. You can have either a fast-moving beam, which allows complex shapes to be drawn without flicker but which tends to produce distortion and undue oscillations at sharp corner points, or a slow-moving beam, which will only work effectively for simpler shapes, but which minimises distortion. As always, a compromise is usually best.

When producing DXF drawings, try to stick to a few simple shapes, using basic primitives. The following restrictions apply;-

- Blocks and groups are not understood. Reduce everything to basic primitives.
- Only the following DXF entities are supported;-
  - LINE
  - POLYLINE
  - CIRCLE
  - ARC
  - 3DFACE

However, since most other shapes can be decomposed to these this is unlikely to prove too much of a handicap.

Scale and positioning are unimportant, since these will all be normalised on import so that the shape fits the screen and with other shapes.

Tweaking performance

Performing 3D and 2D transformations, updating the GUI display, and outputting data in real time to the sound system can prove rather processor intensive. To cope with this, there are several options in the Performance options dialogue to reduce CPU load.

Most significantly, turning off real time update of the 10 bank screens can reduce CPU load by more than half. Also, reducing the frame rate, and turning off anti-aliasing (which only affects the GUI display in any case - you can't anti-alias a vector display!) can also help.

Chapter 2. ScopeShapes menus

File menu

Note

The Mac OS X version has no file menu - the Quit option is on the ScopeShapes menu (as is standard on OS X).

Quit

Quits ScopeShapes.

Options menu

Hardware options

Controls output to the soundcard.

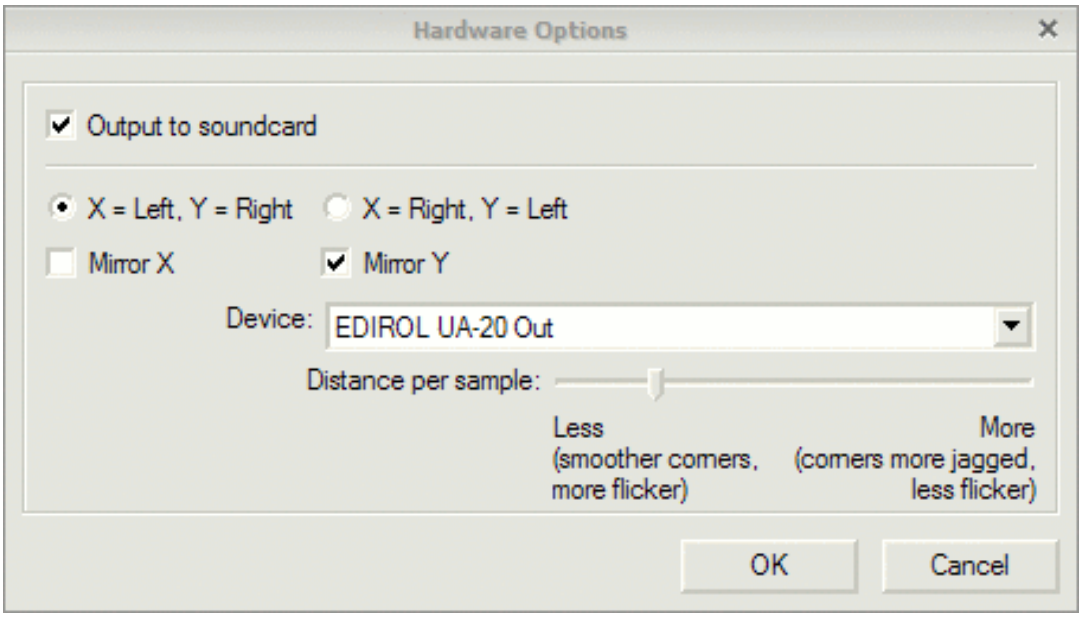


Table 2.1. Hardware options

Output to soundcard	Turns on and off the output to the sound system. Saves a lot of CPU time if turned off, but then that would somewhat defeat the purpose of the whole program.
X = Left, Y = Right	Changes which side of the stereo output drives which axis of the scope. Useful if you've got your wires crossed (although of course you could just swap the connectors on the scope).
X = Right, Y = Left	
Mirror X	Mirrors the signal for either axis. My system seems to need the Y axis reversed for some unknown reason.
Mirror Y	
Device	Selects which audio device to use, if there is more than

	one available. A device will have to have at least 2 output channels to be available.
Distance per sample	As explained in the section 'Importing DXF files', there is a trade-off between distortion introduced by outputting a signal beyond the bandwidth of the audio output, and introducing too much flicker in the display by drawing the trace too slowly. This slider allows a compromise position to be selected.

3D options

Controls the parameters of the 3D display.

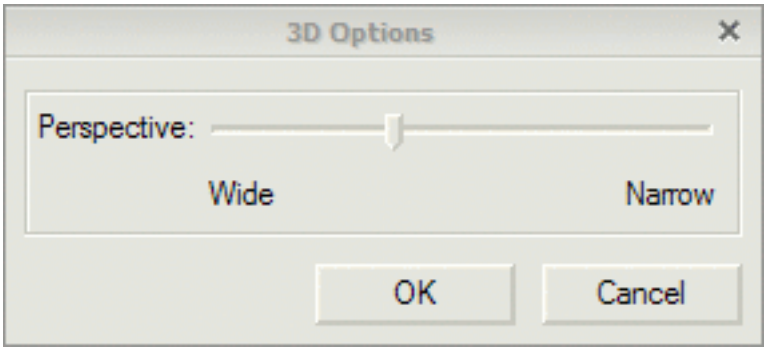


Table 2.2. 3D options

Perspective	Selects between a wide perspective (more dramatically foreshortened lines, like looking through a fish-eye lens), and narrow perspective (flatter-looking, closer to an isometric drawing).

Performance options

Turns on and off features which affect the CPU load.

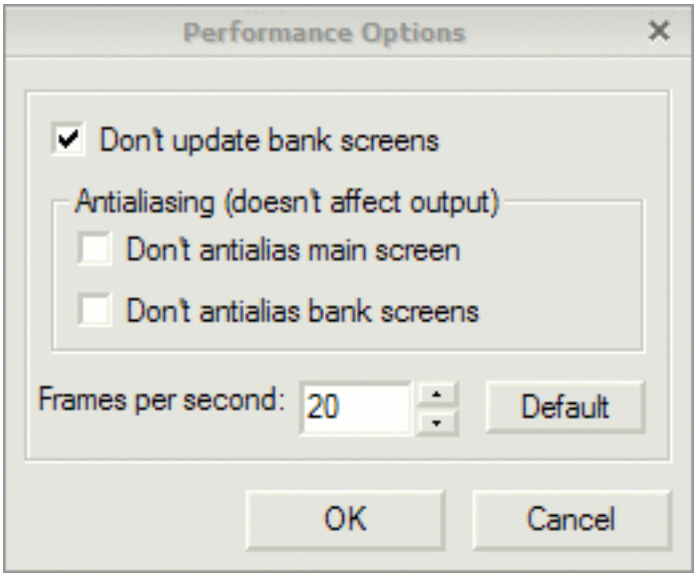


Table 2.3. Performance options

Don't update bank screens	The most significant way to reduce CPU load - as such it is enabled by default. When this option is selected, the display on each bank screen will only be updated when a new shape is loaded. If not, they are all updated in real time.
Don't anti alias main screen	Turns off anti-aliasing (smoothing of lines) for the main screen.
Don't anti alias bank screens	Turns off anti-aliasing for the bank screens.
Frames per second	Selects how often the display will be updated. Lower frame rates consume less CPU power, but make the display more jerky, whilst higher rates produce a smoother image, but consume more CPU power.
Default frames per second	Selects 20 FPS.

Save options on exit

If checked, all the options selected will be saved when the program quits, and will be restored when it is next loaded.

Save options

Saves the currently selected options, so that the next time the program is run the same setup will be in operation.

Help menu

Help contents

Launches the interactive help system.

About ScopeShapes

Note

The Mac OS X version has no about option - it is on the ScopeShapes menu (as is standard on OS X).

Displays some basic information about ScopeShapes.

Chapter 3. Keyboard shortcuts

The following keys can be used in ScopeShapes;-

Table 3.1. Keyboard shortcuts

1 - 9, 0	Switches into a bank. Banks are numbered from the top left, and follow down in rows
Z and X	Adjust transition speed up or down
Up and Down arrows	Adjust X axis rotation speed (these keys are not the wrong way round, since rotation about he X axis means movement in the vertical direction).
Left and Right arrows	Adjust Y axis rotation speed.
Page Up and Page Down	Zoom in and out.

Chapter 4. Credits

ScopeShapes was written by Chris Meighan.

It is written in C++ and may be compiled using GCC [http://gcc.gnu.org/] on Linux and Mac OS X, and MinGW [http://www.mingw.org/] on Microsoft Windows.

The GUI was built using the wxWidgets toolkit [http://www.wxwidgets.com/].

The sound output code makes use of the PortAudio library [http://www.portaudio.com/].

The installer for the Windows version was compiled using Inno Setup [http://www.jrsoftware.org/].

The documentation was compiled using the DocBook system [httpL//www.docbook.org/] with xslt-proc [http://xmlsoft.org/]. The HTML Help version was compiled using Microsoft HTML Help Workshop command line tools [http:://msdn.microsoft.com/], and Apache FOP [http://xml.apache.org/fop/] was used to generate the PDF version.