User Guide

Contents

What is it for?

PhotoChrome started life in the early 1990s as a software-only high colour display system for the Atari ST/E range. Images could be converted into a custom display format which exceeds the display limit of 16 distinct colours from a palette of 512 (or 4096) - which the ST/E hardware normally provides.

Using this method, the standard ST palette range is extended to 3375 colours and STE hardware extends this further to 29791 colours. *9568 distinct colour changes take place within the original PCS display but the number of perceived colours can be significantly higher due to mixing effects.*

Version 5 is a complete rewrite with many new features. It focuses entirely on image conversion. **Display technique is left 'open' for retro hackers, demo coders and homebrew game developers to play with.**

Why another version?

The earlier versions (prior to v5) applied basic conversion techniques in order to run natively on 8MHz ST machines with very quick turnaround. These compromises imposed limits on image quality and degree of control. Images were fixed at 320x199 pixels and the display format was also fixed.

Version 5 (PCS5) has been rewritten with emphasis on **image quality, flexibility and control**. Image conversion can be batched, and it has even been used to produce video streams (see ppera's CATA video system!). A range of dithering and colour error management features have been introduced to make the tool more flexible.

While format compatibility with v4 has been kept, v5 can generate image data for almost *any* 'multi palette' display – including display methods which have not yet been implemented and display sizes larger than 320x199. It can be fed conversion templates in text format which guide the process. The code has been completely rewritten in C++ with a modular layout and has and made public to encourage use and extensions.

What do I need to use it?

PCS5 is intended to be used primarily from a Windows, Linux or Mac environment but can still run on fast Atari systems equipped with FPU. **A number of viewers are available on Atari and PC capable of viewing PCS files. This includes the Atari viewers & code originally released with v3 & v4.**

The tool is driven from command line input. This manual documents the commands and features and provides some examples of use and what to expect. **A built-in help summary can always be summoned using the '–help' command!**

QuickStart: Setup / Usage

PCS5 is a Posix application. This means it is written primarily for a *nix (Linux, Mac) environment but it also operates in a Windows environment (via Cygwin – a kind of Linux emulation layer).

Windows note: It is not necessary to have the Cygwin software installed to use PCS5 – a few DLL files have been provided along with the executable to allow the tool to run independently and these should be located next to pcs5.exe or in a system search path – but if you do have Cygwin installed it is recommended you ignore the supplied DLLs in favour of the installed version!

The Atari version is supplied as TTP files and a 68020-class CPU and floating point unit (FPU) is required to run these. A 68000 version can be compiled but expect very long waits...

There are no other prerequisites for **using** the tool. It operates from the command line.

*If you are using a different kind of operating system the tool can be compiled from source. There are prerequisites for **compiling** the tool (the FreeImage library) so it is up to you to locate and compile that material beforehand if you decide to build your own executable.*

Basic usage pattern...

From DOS prompt:

```
pcs5 [options] [-o outfile.pcs] infile(s)
```

Or from the Cygwin bash shell:

```
./pcs5.exe [options] [-o outfile.pcs] infile(s)
```

The output file will default to INFILE.PCS but can be overloaded with '–o'.

If you specify infile as a GLOB for batching (e.g. *.PNG) overloading should be avoided – each output file will be named according to the corresponding input file.

Don't enable diagnostic mode when batching – it will just overwrite the same diagnostic files over and over for each image. At some point the diagnostic images will be filed in folders to make this more useful with batching.

QuickStart: Test - STE Output

For quick results on an STE from a 29791 colour palette, try one of the sampling reduction methods e.g. –m 4:

```
pcs5 –cd ste –f 2 –m 4 –lt 3 –dt 0 –et 2 test.jpg
```

If you don't mind waiting and want to **concentrate on image quality**, or if the image is causing difficulties with high colour variation, try the neural network colour reducer in dual-field mode at a refinement depth of 12 (4096 iterations) with 4 passes of field error diffusion.... And make a cup of tea or go for a walk.

```
pcs5 –cd ste –f 2 –m 5 –nnd 12 –lt 3 –dt 0 –et 1 –ei 4 test.png
```

Some **less challenging images** can look more accurate and stable (and produce smaller files) using only palette interlacing ('shared bitmap' mode). Sometimes it's worth trying this because when it works it can produce the most stable display. Files are smaller because only one bitmap generated and displayed, and both palette fields are very similar. *This mode is not as powerful as dual-field mode for dealing with colour variation and shading – look out for visible streaks and patches.*

```
pcs5 –cd ste –f 1 –fm 1 –m 5 –lt 3 –dt 0 –et 2 test.png
```

And for 100% stability (but without the extended palette – 4096 colours only), you can use this to generate a single-field image:

```
pcs5 –cd ste –f 1 –m 5 –dt 0 –et 2 test.png
```

QuickStart: Test - ST(FM) Output

The ST(FM) equivalents for the above cases is the same – just pass '-cd st' instead of '-cd ste' to generate the appropriate colour depth. Due to the reduced colour depth, some settings will be more or less appropriate for ST vs STE. E.G. some types of dithering look better at higher colour depths while others suit lower colour depths. These are all explained later.