WarpJPEG

Oliver Roberts

COLLABORATORS

	TITLE : WarpJPEG	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Oliver Roberts	August 27, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

WarpJPEG

1.1 WarpJPEG.datatype 44.5

```
                WarpJPEG.datatype 44.5 - the
           fastest
            24-bit JPEG picture datatype!
                              (for WarpUp PPC systems only)

      Copyright © 1999-2000
             Oliver Roberts
             , All Rights Reserved.



              Description
              what is this datatype for?

              System requirements
              what you need to use this software

              Installation
              installing this software

              Speed
              information regarding speed issues


              Distribution
              distribution conditions

              Disclaimer
              important notices


              Acknowledgements
              thankyous and credits

              About the author
              how to contact the author

              Future
```

1.2 Description

As you've probably guessed, WarpJPEG.datatype is a yet another JFIF/JPEG datatype. The difference is that this is for owners of PPC cards, and it has been targetted specifically at WarpUp (not PowerUp). Even better, it is fast, compact, clean, well behaved and fast - a true plug'n'play PPC datatype.

I decided to write this datatype because no other decent WarpOS native JPEG datatype exists for the PPC, and I thought this would be the ideal chance to finally learn how to start coding PPC stuff on the Amiga. Yes, this is my first PPC project, although I have worked a lot with JPEG coding and datatype coding already, so I do know what I'm doing, honest ;)

1.3 System Requirements

This datatype needs the following in order to work:

- PPC accelerator card + 68040/060
- WarpUp Release 4 (powerpc.library V15) or higher
- picture.datatype v43 or higher
  (i.e. either of the ones supplied with AmigaOS 3.5, P96 or CGraphX)
- Kickstart 3.0 or higher

Note that a graphics card is not necessary - the P96 or OS 3.5 picture.datatype will automatically dither images down to your native Amiga display requirements.

1.4 Installation

To install this datatype, simply run the provided installer script by double-clicking the icon.

Alternatively, the datatype can be installed manually (be sure to understand the requirements first though) by typing the following shell commands:

```
Copy "Devs/Datatypes/JPEG#?" DEVS:Datatypes
Copy Classes/Datatypes/WarpJPEG.datatype SYS:Classes/Datatypes
AddDataTypes REFRESH
```

You'll probably want to make a backup of your existing JPEG descriptor (the

file in DEVS:Datatypes) first. You will also need to ensure there are no other JPEG/JFIF datatypes in DEVS:Datatypes. Finally, a reboot may be needed for the changes to take effect.

1.5 Speed

The fastest JPEG datatype?

WarpJPEG is currently the fastest JPEG datatype available. The table below shows the time (in seconds) it took to decode 10 different images on my A1200 603e/240MHz 060/50MHz, with BVision and CGX picture.datatype, with other tested datatypes configured as close as possible to WarpJPEG's internal settings:

	PowerPC		M68K	
	WarpJPEG 44.4	akJFIF 44.65 WarpUp / 68K	JFIFdt G.Nikl	OS 3.5
2008x1597 col, 225570 bytes	2.13	3.66 6.25	8.16	8.80
1280x1012 col, 148387 bytes	0.96	1.70 3.05	3.72	3.96
804x1040 col, 306821 bytes	1.01	1.58 3.67	3.45	3.54
1024x 766 col, 244522 bytes	1.05	1.55 3.53	3.29	3.33
779x 767 col, 162752 bytes	0.71	1.12 2.55	2.30	2.35
779x 767 col p, 86016 bytes	1.20	1.63 3.28	2.92	3.05
450x 450 col, 50845 bytes	0.27	0.55 0.99	0.77	0.78
370x 502 col, 24480 bytes	0.19	0.43 0.69	0.60	0.63
506x1007 gry, 88115 bytes	0.27	0.55 1.08	1.16	0.78
761x 495 gry p, 68593 bytes	0.42	0.65 1.35	1.25	1.05

(p = progressive)

These figures are provided for comparison purposes only, but they speak for themselves... WarpJPEG is around 60-70% faster on most images (twice as fast on greyscale and small colour images) than the native WarpOS version of akJFIF and around 3 times faster than 68k datatypes on a 060 (the speed difference will be even greater on 040 systems with a fast PPC). All this and no loss in image quality!

All tests were performed, multiple times, using Visage, with the following command line: "visage test.jpg nojpeg time test".

Still too slow!

Despite these facts, I'm still disappointed with the relatively slow performance advantage offered by my 603e over my 060 (approximately only 3 times faster). The main problem is that PPC datatypes still have to use the 68k for reading the data from disk and for creating / writing to the bitmap structures that the datatypes system requires. As far as the former goes, time lost for file i/o is negligable as WarpJPEG uses double buffered asynchronous i/o (supports DMA controllers).

The largest bottleneck is that the DTM_WRITEPIXELARRAY method of the picture.datatype has to be used to write the image data from WarpJPEG into

the image bitmap. As this process is done via picture.datatype, it can only currently be performed by the 68k. To give you some idea of how much of a problem this is for WarpJPEG, typically, half of the overall decode time is used by the PPC to decode the whole image, and the other half is used by DTM_WRITEPIXELARRAY on the 68k. And that's on a graphics card - the time used by DTM_WRITEPIXELARRAY will probably be even greater on systems using native Amiga graphics. It doesn't take a genius to see that this is slowing the datatype down, and is the main reason why WarpJPEG will still be faster on a 060 than a 040.

How to make the datatype faster

Is there anything that can be done about this? Well, yes, there are a few patches that you can install which should make things faster:

- NewWPA8 (util/boot/NewWPA8.lha on Aminet) should provide a notable speed increase on native Amiga graphics - probably won't make any difference if you use a graphics card.

- If you use a graphics card and CyberGraphX, you may want to make sure you are using the supplied v43 picture.datatype, as this will be faster than the P96 and OS3.5 picture.datatype on your system.

Of course, any other general speed-up patches should help too.

1.6 Distribution Conditions

WarpJPEG.datatype is public domain with the copyright remaining ↩
with the

author
and may be freely distributed legally providing:

(1) None of the distributed files are changed in any way
(2) It is not sold for profit and it is not included on any disks
that are sold solely for profit (includes magazine coverdisks)
(3) The distribution contents remain complete (see list below)

If this software is to be sold for profit, permission must be obtained from me, the
author
.

Aminet, Amiga Format and Amigactive have been granted permission to distribute WarpJPEG.datatype on their CDs.

The following files must be present in their original and unchanged form in any copies of this software:

Classes/Datatypes/WarpJPEG.datatype
Devs/Datatypes/JPEG
Devs/Datatypes/JPEG.info
WarpJPEG.guide
WarpJPEG.guide.info
Install_WarpJPEG

```
   Install_WarpJPEG.info
```

1.7 Disclaimer

```
This software is provided "as is", without warranty of any kind, either
expressed or implied, statutory or otherwise. By using the archive and its
contents, you accept the entire risk as to its quality and performance.

Neither Oliver Roberts nor any other party involved in the creation,
production or delivery of the archive and its contents shall be liable for
any direct, indirect, special, consequential or incidental damages,
including without limitation damages for loss of profits, loss of use or
loss of anticipated costs, expenses or damages, and any data or information
which may be lost or rendered inaccurate, even if Oliver Roberts is advised
of the possibility of such damages.

Do not attempt to tamper with the supplied files. Doing so will cause
problems and you may find things start going wrong!
```

1.8 Acknowledgements

```
This software is based in part on the work of the Independent JPEG Group,
using the libjpeg link library (compiled for WarpOS).

It was made possible by VBCC, which was used to build and compile the
datatype.  Thanks to Volker Barthelmann and the other authors involved.

Thanks also to Sam Jordan for WarpOS and helping me out with various
queries regarding it.

Finally, thanks to the OS 3.5 development team - now everyone has access to
a 24-bit picture.datatype, I don't need to bother messing about adding
dithering routines :)
```

1.9 About the author

```
If you have any problems with this software, or if you have any
suggestions/queries, please contact me and I will do my best to sort any
bugs out as soon as possible:

e-mail: oliver.roberts@iname.com
   www: http://www.nanunanu.org/~oliver/
   icq: 34640231
```

1.10 Future

```
                Some things that may appear in the future:

 - If I can squeeze any more speed out of the datatype, I'll do so :)

If you have any other suggestions, please
                let me know
                .

Future releases of WarpJPEG.datatype will be available from either Aminet
(util/dtype/WarpJPEGdt.lha) or its webpage:

  http://www.nanunanu.org/~oliver/warpjpeg.html
```

1.11 Program History

```
                44.5 (9.1.2000)

 - When the OS3.5 picture.datatype is in use, dithering is now switched
   off (only) if the image is to be rendered to a hi/true colour screen,
   resulting in much higher performance, with negligable quality loss.
 - Enhanced descriptor file to detect EXIF and Photoshop JPEG file
   variations.
 - Fixed bug in error handling.
 - Dispatcher now performs some extra functions, which may quash the
   stability problems that some users have experienced.

44.4 (5.1.2000)

 - Fixed silly bug new to v44.3 which caused a lot of images to be
   rendered with half the vertical resolution (i.e. every other line was
   left out and the rest were doubled), resulting in blocky images.
 - Ironically, fixing the above bug enabled the optimizations that I had
   originally intended: 30-50% faster for colour images, and around 100%
   faster for greyscale images!
 - Reverted to my simpler and less hungry asynchronous engine, as the
   new engine in 44.3 no longer offers an advantage with the above
   optimizations in place. [after release this was found to solve major
   stability problems that some users experienced with v44.3]
 - Corrected akJFIF WarpUp
                benchmarks
                .

44.3 (3.1.2000)

 - At least 35% faster overall (upto 60% faster in many cases!), with
   no loss in the output image quality.
 - Enhanced descriptor file to detect more JPEG file formats.
 - Now uses asynchronous i/o routines which has made things a little bit
   faster overall, but will be most noticable on large (low compression)
   files and/or slower media.
 - Improved error handling.
 - Added
                Speed
                 section to this documentation.
```

44.2 (18.12.1999)

- Now at least 10-25% faster (depending on the JPEG file) after optimizations and tweaking libjpeg settings.
- Greyscale jpegs are now displayed if using the OS 3.5 or P96 picture.datatype (only worked with the CGX one in v44.1).
- Fixed little bug in the lib init (wasn't setting the revision number).

44.1 (15.12.1999)

- Initial release.