NAME
warp - apply a uniform shift or non-uniform shift (warp) to an image

SYNOPSIS
warp image-file shift-file out-file [ -q ] ...

DESCRIPTION
warp shifts a 2-D image, f, read in from image-file by shift amplitudes
delx, dely, read in from shift-file and computes a shifted image, fs,
which is written to out-file.

The image, f, can be written to image-file with the IDL procedure vcimage\negelout.pro. Files written with vcimage2out.pro, or vcimage3out.pro
 can be used as well, but warp will only read the 1st image from the
file.

The shift amplitudes, delx and dely, can be either scalars, in which
case the shifts are uniform, or they can be 2d arrays with the same
dimensions as the image f, in which case the shifts are non-uniform
(warping). The units of delx, dely are assumed to be in pixel units.
The shifts delx, dely can be written with the IDL procedure vcimage\negelout.pro.

After the input image has been shifted or warped, and written into out-
file it can be read into an IDL session with the procedure vcimage\nelin.pro .

All of these IDL procedures are in the IDL-io-procedures folder. The
data in image-file, shift-file, and out-file are stored in binary,
large-endian byte order, and warp and the IDL I/O procedures to read
and write the files should be platform independent.

The warp program has the same functionality as the shift_frac2d.pro IDL
procedure, but is considerably faster when performing image warping.
The mathematical technique is based on the fact that shifting an image
can be viewed as the convolving the image with a delta function at the
desired shift. The Fourier transform of the delta function can be
written analytically, allowing one to compute the shifted image using
standard Fourier transform techniques.

OPTIONS
-q If this flag is set, no non-error output is sent to stdout.

EXAMPLES
example warping an image in an IDL session:

IDL>f1=randomu(seed,201,101)
IDL>vcimage\negelout,f1,’imagefile.dat’
IDL>delx=cos(!pi*findgen(201)/200.)#sin(!pi*findgen(101)/100.)
IDL>dely=sin(!pi*findgen(201)/200.)#cos(!pi*findgen(101)/100.)
IDL>vcimage\negelout,delx,dely,’shiftfile.dat’
IDL>$warp imagefile.dat shiftfile.dat outfile.dat
IDL>vcimage\nelin,f2,’outfile.dat’
example applying a uniform, non-integer shift to an image in an IDL session:

    IDL>f1=randomu(seed,201,101)
    IDL>vcimage1out,f1,'imagefile.dat'
    IDL>delx=1.5
    IDL>dely=-0.5
    IDL>vcimage2out,delx,dely,'shiftfile.dat'
    IDL>$warp imagefile.dat shiftfile.dat outfile.dat
    IDL>vcimagelin,f2,'outfile.dat'

Print out short summary of documentation:

    warp

FILES

There are no configuration files.

AUTHORS

George H. Fisher, SSL UC Berkeley <fisher at ssl dot berkeley dot edu>
Brian T. Welsch, SSL UC Berkeley <welsch at ssl dot berkeley dot edu>

SEE ALSO

source code of vcimagelin.pro (IDL procedure), source code of vcimage2out.pro (IDL procedure), and source code of vcimagelin.pro (IDL procedure), and source code of shift_frac2d.pro (IDL procedure).