# SIM_HOL - An implementation of the angular spectrum propagation method for reconstructing holograms designed using DS_EXPT

Matt Clark

March 4, 2005

## 1 Release notes

This is sim_hol (c) Matt Clark September 2002 This release is covered by GPL. The license statement is located at the end of this document in section 5 and full details were included in this distribution in a file called COPYING.

### 1.1 Revision history

1998 sim_ap version 4.5
2002 sim_hol pared version 1.0 GPL public release.

## 2 Introduction

SIM_HOL is a program which can simulate the reconstruction of holograms. Specifically it uses angular spectrum propagation to project the complex amplitude (optical wave) which results from the hologram across free space to an image plane. It contains a few handy features useful for simulating holograms designed using the DS_EXPT direct search hologram design program. In particular this includes loading of .hol files formats and automatic scaling or transformation of the reconstruction.

## 3 Building the software

### 3.1 FFT

This program uses FFTs to go to and from the angular spectrum representation of the wavefield. This implementation uses the excellent and publicly available FFTW library. This implementation also exclusively uses the double precision library which on my system is installed as libdfftw and uses the header files dfftw.h. If you only have double precision libraries installed you may have to use libfftw and fftw.h (note the missing d).

### 3.1.1 Getting FFTW

You can obtain fftw from `http://www.fftw.org` or it may have been bundled with this software. It is (in my experience) simple, easy and reliable to build. Recently I have been using an rpm or deb file to install fftw - this seems to work with the linking with no additional effort.

You must use fftw version 3 or above.

### 3.1.2 Using other FFT packages

You can easily use another FFT routine by hacking the source code. I can't see why you would bother though, FFTW is generally pretty fast in all circumstances. Obtaining it and building it is easier than the trivial source code hacks required to use an alternative.

## 3.2 Making the software

Once you have a FFTW the next steps should be easy, unpack the source code and then use "make" to build it. If you have put FFTW in a non-standard place you may have to point the make file to the headers and libraries.

The only build option is to supply a filename for the storage of FFTW wisdom files. The default is "/tmp/sim_hol.wisdom" which isn't such a bad place if you aren't on a shared machine. this file should really be deleted everytime you recompile - you can do this manually.

# 4 Using the software

SIM_HOL takes one argument, the name of the control file. Traditionally this name ends with .sim. A typical .sim file looks like this,

```
inname  gasket_gs.hol
outname gasket_gs.dat
focal_length    0.5
psize   10e-6;
pixels  512 512
autoscale
phase 2
```

two other commands are supported in this version: `N <xx>` sets the simulation size overriding the autoscaling choice and `F_bar <ff>` sets the "focal length" of the zoom / scaling maths, a setting of 0 indicates infinity. `F_bar` defaults to 0 (infinity) without autoscaling enabled and is set by the program if autoscaling is enabled. If autoscaling is enabled then the value supplied for the focal_length ($F$) is overridden by $F'$, otherwise the reconstruction takes place at $F$. `inname` sets the input filename, `outname` sets the output filename, `psize` sets the pixel size of the input, `pixels` sets the number of pixels in the hologram, `autoscale` enables autoscaling and `phase` sets the number of phase levels used in the hologram.

The input format is binary signed chars for the .hol files and binary double precision C++ complex numbers for the output.

## 4.1 Autoscaling

In most hologram designs the spatial size of the image is far larger than the spatial size of the hologram. The hologram must be sample with at least one simulation pixel per hologram pixel. The number of pixels (at this size) required must be adequate to contain the image (otherwise you get aliasing). Without scaling meeting these two conditions can require extremely large arrays.

With the scaling transform the effective size of the pixels can be transformed during propagation so all you need is an array size that adequately samples the hologram. If left to the program the number of (linear) pixels used is the next power of two greater than the number of (linear) pixels in the hologram. The pixel size is then scaled to so that this array fits the image. The image size is computed by considering the critical first order diffraction angle of the hologram. This is a little conservative so watch out.

For a technical explanation of how this works read the next section otherwise be content that this feature saves you lots of hassle.

### 4.1.1 Autoscaling - technical

Despite what I said in the above section it is not possible to transform the pixel size - this remains constant. Instead what is done is to transform the field of view, this is performed with the mathematical equivalent of a zoom lens. The image (with the addition of the zoom lens) is refocused from the original plane at $F$ to $F'$ the propagation then takes place not to $F$ but to $F'$. This transforms the field of view by a factor $F'/F$ which is equivalent to transforming the pixel size to $F/F'$.

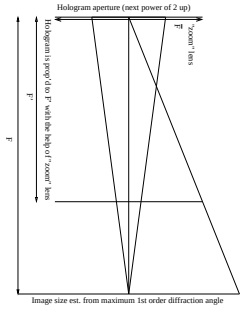


Figure 1: Schematic showing how autoscaling works

You can set your own scaling by not specifying autoscale and setting $N$, $\bar{F}$ (F_bar) and $F$ manually (in this case F actually sets $F'$ the actual propagation distance). You can calculate $\bar{F}$ using simple thin lens formulae.

# 5 License - GPL

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# References