NASA/TM–2017-219664

Rapid Generation of Large Dimension Photon Sieve Designs

Shravan Hariharan
Georgia Institute of Technology, Atlanta, Georgia

Sean Fitzpatrick
Georgia Institute of Technology, Atlanta, Georgia

Hyun Jung Kim
National Institute of Aerospace, Hampton, Virginia

Matthew Julian
University of Virginia, Charlottesville, Virginia

Wenbo Sun
Science Systems and Applications, Inc., Hampton, Virginia

Ken Tedjojuwono and David MacDonnell
Langley Research Center, Hampton, Virginia

September 2017

NASA/TM–2017-219664

Rapid Generation of Large Dimension Photon Sieve Designs

Shravan Hariharan
Georgia Institute of Technology, Atlanta, Georgia

Sean Fitzpatrick
Georgia Institute of Technology, Atlanta, Georgia

Hyun Jung Kim
National Institute of Aerospace, Hampton, Virginia

Matthew Julian
University of Virginia, Charlottesville, Virginia

Wenbo Sun
Science Systems and Applications, Inc., Hampton, Virginia

Ken Tedjojuwono and David MacDonnell
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

September 2017

Acknowledgments

This research was fulfilled in the Class100 (ISO5) Optical Cleanroom at AMDSB (Advanced Measuremetns and Data Systems Branch) at NASA Lanlgey Research Center. This research was supported by the NASA Langley Research Center Internal Research and Development (IRAD) Program, which is designed to develop strategic technological capabilities within the center. We would like to thank Scott Bartram for the video recording and the photon sieve project team. In addition, we would like to thank the NASA Internships, Fellowships, and Scholarships (NIFS) Program, which supported the selection and funding of the interns (Shravan Hariharan and Sean Fitzpatrick) who contributed to the project. Last but not least, we would like to thank Thomas Jones (AMDSB head) and Robert Bryant (Advanced Materials and Processing Branch head) who provided valuable insight and assistance during the writing of this report.

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available from:

NASA STI Program / Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199
Fax: 757-864-6500

Abstract

A photon sieve is a revolutionary optical instrument that provides high resolution imaging at a fraction of the weight of typical telescopes (areal density of 0.3 kg/m^2 compared to 25 kg/m^2 for the James Webb Space Telescope). The photon sieve is a variation of a Fresnel Zone Plate consisting of many small holes spread out in a ring-like pattern, which focuses light of a specific wavelength by diffraction. The team at NASA Langley Research Center has produced a variety of small photon sieves for testing. However, it is necessary to increase both the scale and rate of production, as a single sieve previously took multiple weeks to design and fabricate. This report details the different methods used in producing photon sieve designs in two file formats: CIF and DXF. The difference between these methods, and the two file formats were compared, to determine the most efficient design process. Finally, a step-by-step sieve design and fabrication process was described. The design files can be generated in both formats using an editing tool such as Microsoft Excel. However, an approach using a MATLAB program reduced the computing time of the designs and increased the ability of the user to generate large photon sieve designs. Although the CIF generation process was deemed the most efficient, the design techniques for both file types have been proven to generate complete photon sieves that can be used for scientific applications

Table of Contents

List of Figures

List of Tables

Nomenclature

LaRC	NASA Langley Research Center
PS	Photon Sieve
FZP	Fresnel Zone Plate
DXF	Drawing Interchange Format
CIF	Caltech Intermediate Format
SCR	AutoCAD Script File

1. Introduction

A photon sieve is a revolutionary optical instrument that uses diffraction, rather than reflection or refraction as most telescopes do [1]. Reflective and refractive telescopes contain heavy glass lenses or mirrors, which are costly to use in space applications. Consisting of many small pinholes spread out in a ring-like pattern, the photon sieve (PS) is a variation of a Fresnel Zone Plate (FZP), which focuses light of a specific wavelength. Due to the interference of light rays passing through the rings, a sharp image is created at the focal point, which can be detected by a camera or beam profiler. However, FZP's are limited in spatial resolution by the width of the outermost zone, and often create unwanted diffraction orders and additional scattered intensity. By using a photon sieve, the focal spot can be sharpened and the FZP limitations can be removed by varying the pinhole diameters and distribution, which improves the imaging quality. By using a flexible membrane photon sieve (areal density of 0.3 kg/m^2 compared to 25 kg/m^2 for the James Webb Space Telescope) on a space telescope, the weight of the mission is reduced [2]. This reduces the cost while maintaining a high imaging quality. Missions using the photon sieve for space applications are under development [2], but no such mission has been launched to date. The team at NASA Langley Research Center (NASA LaRC) has produced a variety of small photon sieves for testing, but it is necessary to increase both the scale and rate of production, as a single sieve previously took multiple weeks to design and fabricate. After the process of preparing and manufacturing large glass photon sieves is optimized, the team can proceed to generate large membrane photon sieves for space applications.

2. Project Goal

The prototype photon sieve will be made out of glass, with a target diameter requirement of six inches (152.4 mm). For extremely large files, with millions or even billions of holes, an optimal method must be developed to streamline this process and reduce computing time so that it does not take weeks to generate large sieves. In addition, the files generated must be compatible with the Heidelberg DWL 66fs laser lithography system [3], and the team must also be able to preview the files for errors prior to manufacturing. The sieve will then be manufactured and tested in a laboratory environment. This report describes the two file formats used, with extensions CIF and DXF. In addition, the various methods to generate and test these files are outlined.

3. Heidelberg DWL 66fs System

The manufacturing system used throughout this process was the Heidelberg Instruments DWL 66fs laser lithography system. This instrument is compatible with four file formats: DXF, Gerber, GDSII, and CIF [3]. For the photon sieve design process, only the DXF and CIF formats were used, and therefore are the only formats discussed in this report. Both are human-readable file formats, consisting of ASCII values. In addition, unlike the Gerber and GDSII formats, both DXF and CIF easily support circles and are designed for simple manual file writing. In addition to the format-specific rules described later in the report, Heidelberg Instruments released a series of common design rules to follow when generating the files. These rules are explained in further detail in the Heidelberg DWL 66fs user guide, which is found in the References section.

Common Rules:

- All Polylines should be closed. Although some data formats allow automatic closing of polylines, be aware of the possible ambiguities of not properly closed polylines.
- All closed polylines will be completely filled out on the inside. If there is a frame around an object, it must not shadow the structures inside. This can be avoided by using the XOR mode feature.
- Polylines crossing themselves will produce data errors.
- "Doubled" vertices in polylines (which means the same coordinates are used twice without any other point between them) will produce data errors.
- Single lines with no width will be ignored (best case) or will cause the conversion to fail.
- Polygons must have not more than 128000 vertices.
- Definition or reference depth can be at maximum 16.
- The number of definitions or references can be at maximum 8192.
- Text is not supported (except for DXF).
- Designs must not exceed the limits of 2000 mm from (0, 0) in any axis.

4. Use of MATLAB

MATLAB was used to create the automated conversion programs described later in the report. The language was chosen as it was designed specifically for the simple control of large matrix manipulations and complex numerical calculations. As the photon sieve design process involves the manipulation of arrays with thousands of elements, it was imperative that the programming language chosen expedited the handling of these arrays. However, it is possible to implement the design algorithm in other, possibly open-source languages, such as Python. The use of other languages is not discussed in this report. The precise MATLAB programs used are shown in full for reference in Appendix A and Appendix B.

5. CIF File Generation

There are various methods to generate CIF files, which are described in the section below. First, the file format was outlined, along with examples of how the file may be written. Next, the initial manual method of generating CIF files using a FORTRAN output and Microsoft Excel is described. Finally, this report details the MATLAB programs written to directly generate CIF files from a set of given parameters.

5.1 CIF File Format

Caltech Intermediate Format, or CIF, is a format that describes mask layout for manufacturing [4]. This format is essentially a text file, with a header followed by a description of the features involved. The Each feature as well as the format of the file are described by commands, of which six are relevant for photon sieve generation. The commands and their format are detailed below. All units in the CIF format are in centimicrons (10^{-8} meters).

Header:

1) Definition Start (DS # # #;) – Begins the defined pattern
2) Layer Name (L _____;) – Names the layer being drawn

Geometric Features:

3) Roundflash (R # # #;) – Draws a circle of diameter # centered at #,#

Footer:

4) Definition Finish (DF;) – Ends the definition of the above pattern
5) Call Subroutine (C #;) – Calls the pattern to be drawn by the machine
6) End File (E;) – Ends the CIF file

Every command is ended with a semicolon, with the following command beginning on the next line. Below, each command and the necessary syntax are explained in detail.

1) Definition Start
    a. Must be the first line in the CIF file
    b. "DS" Followed by three integers
        i. Subroutine number – this is simply a way of organizing patterns so that they can be called later on (1 for first subroutine, 2 for second, etc.)
        ii. The second and third numbers are the scale factor. The second number is the numerator and the third is the denominator. For example, a design scaled to 1/10th the stated size would have "1" in the second space and "10" in the third. This is useful for subroutines containing features with dimensions much larger or smaller than a few centimicrons.
    c. Example – "DS 1 1 1;" defines subroutine "1" with scaling "1:1"
2) Layer Name
    a. Written as "L" followed by the layer name

    b. Only one layer read by the Heidelberg machine
    c. Each layer requires its own subroutine (DS)
    d. Example: “L 1;” names the layer “1”. L Top; names the layer “Top”
    e. Layer Name should follow Definition Start, on the next line of the CIF file

3) Roundflash
    a. Draws a filled circle
    b. Format is “R diameter x y;” where diameter, x, and y are the respective values
    c. Diameter, x and y must be integer values, unit is centimicron (10^{-8} microns)
    d. Example: “R 10 0 0;” gives a circle of diameter 10 centimicrons centered at (0,0)
    e. Each feature is written as its own line in the CIF file

4) Definition Finish
    a. Written as “DF”
    b. Finishes the subroutine definition (closes the DS)
    c. Immediately follows the last line of geometry definitions

5) Call
    a. Written as “C” followed by the subroutine number
        i. Subroutine number is the first integer in the “DS # # #;” line
    b. If multiple subroutines are present, all must be defined first, and then called in the desired order at the end of the CIF file
    c. Immediately follows the Definition Finish command
    d. Example - “C 1;” calls subroutine “1” to be executed

6) End
    a. Written as “E;”
    b. The last line in the CIF file
    c. Ends the program

In addition to the standard CIF format as well as the common design rules described earlier, Heidelberg Instruments released a series of CIF specific rules to follow to ensure proper file reading.

CIF Specific Rules:

- Definition or reference depth can be at a maximum value of 16.
- The number of definitions or references can be at a maximum value of 4096.

Following these rules, an example CIF file is shown below, with the corresponding commands labeled.

DS 1 1 1; *(Definition Start, Subroutine 1, Scale 1:1)*

L Example*;* *(Layer name is “Example”)*

R 10 0 0; *(Draw a circle with a diameter of 10 centimicrons, centered at (0, 0))*

R 1 5 5; *(Draw a circle with a diameter of 1 centimicron, centered at (5, 5))*

DF; *(Definition Finish)*

C 1; *(Call Subroutine 1)*

E; *(End of file)*

For the Heidelberg DWL 66fs to read the design file, it must be written in the exact format described above, including the use of spaces and semicolons. If the format is incorrect, the instrument will display an error when attempting to read the file. For designs on a scale other than centimicrons, the "Scale" feature of the "Definition Start" command should be used appropriately.

5.2 Manual CIF Conversion

The manual conversion process from the raw FORTRAN data to a readable CIF format is comprised of three main steps, shown below. The following section shows each of these steps in detail.

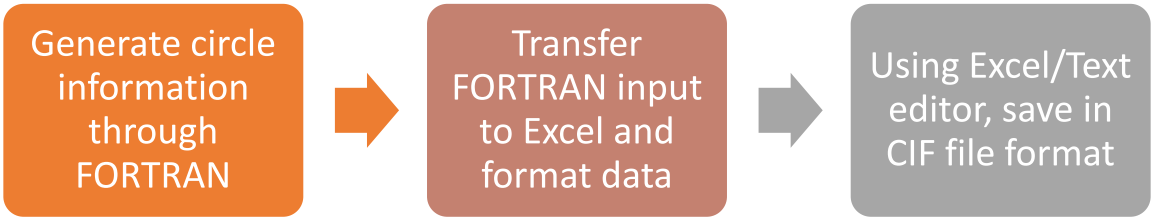

Generate circle information through FORTRAN
Transfer FORTRAN input to Excel and format data
Using Excel/Text editor, save in CIF file format

Figure 1. Manual CIF Conversion Process

3.2.1 FORTRAN Generator

The CIF generator code was written in FORTRAN by Dr. Wenbo Sun. It creates a text file with a header followed by five columns of information dictating the location and size of each hole to be generated. The most recent format has columns in the following arrangement, an example of which can be seen below in Figure 2:

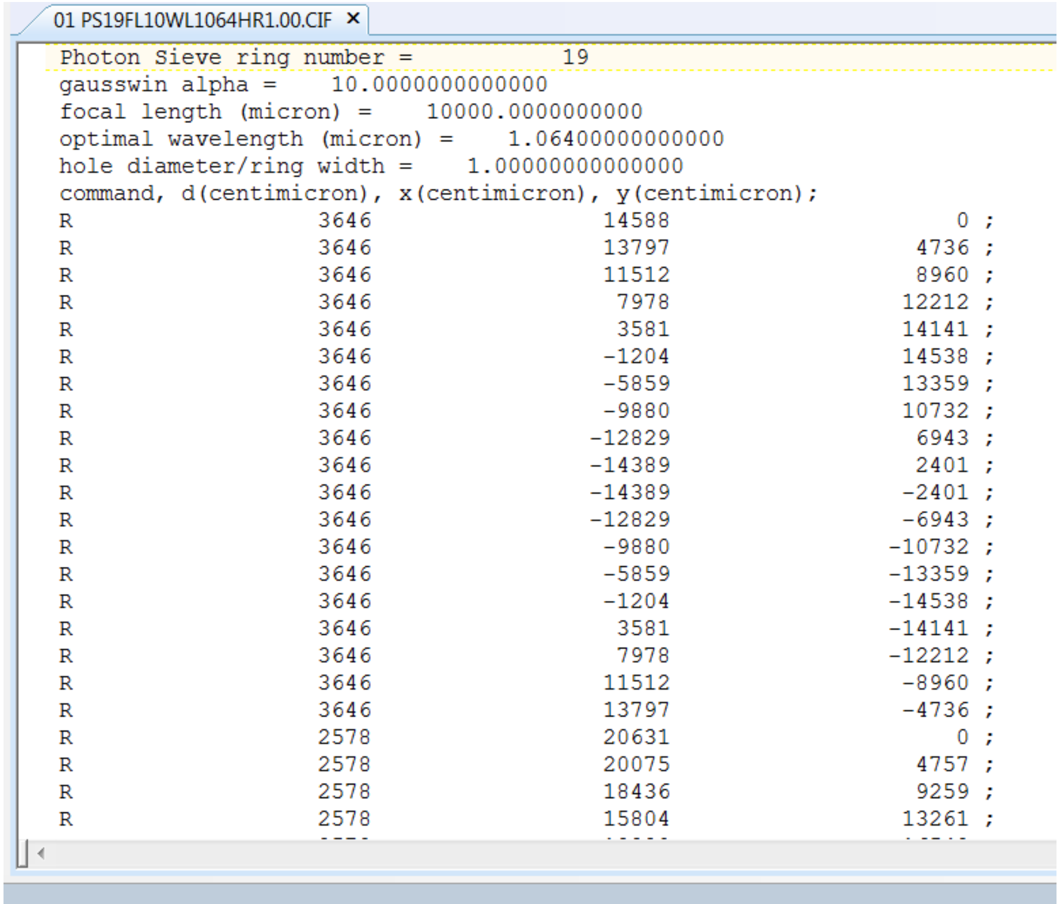
Column one: R's (designation for "circle")

Column two: pinhole diameter (centimicrons)

Column three: pinhole center x-coordinate (centimicrons)

Column four: pinhole center y-coordinate (centimicrons)

Column five: Semicolon



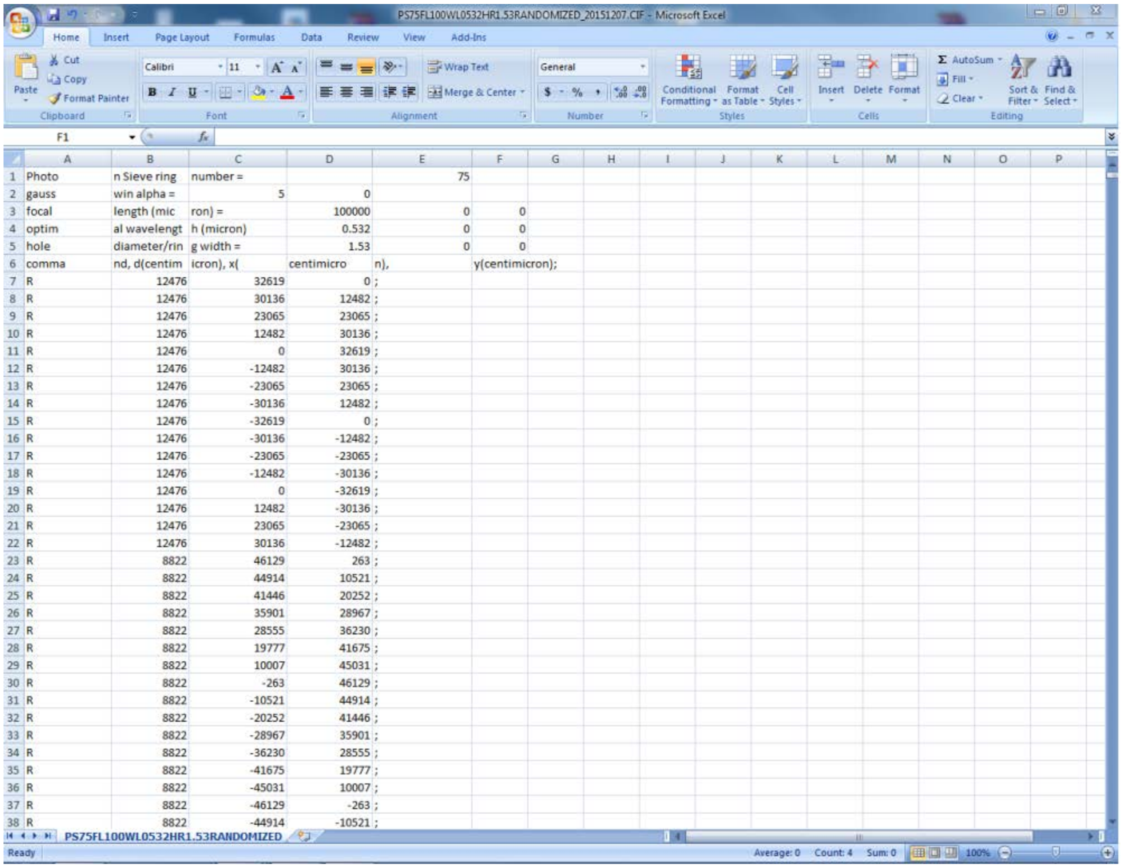
```
Photon Sieve ring number =           19
gausswin alpha =    10.0000000000000
focal length (micron) =    10000.0000000000
optimal wavelength (micron) =    1.06400000000000
hole diameter/ring width =    1.00000000000000
command, d(centimicron), x(centimicron), y(centimicron);
R                  3646                 14588                    0 ;
R                  3646                 13797                 4736 ;
R                  3646                 11512                 8960 ;
R                  3646                  7978                12212 ;
R                  3646                  3581                14141 ;
R                  3646                 -1204                14538 ;
R                  3646                 -5859                13359 ;
R                  3646                 -9880                10732 ;
R                  3646                -12829                 6943 ;
R                  3646                -14389                 2401 ;
R                  3646                -14389                -2401 ;
R                  3646                -12829                -6943 ;
R                  3646                 -9880               -10732 ;
R                  3646                 -5859               -13359 ;
R                  3646                 -1204               -14538 ;
R                  3646                  3581               -14141 ;
R                  3646                  7978               -12212 ;
R                  3646                 11512                -8960 ;
R                  3646                 13797                -4736 ;
R                  2578                 20631                    0 ;
R                  2578                 20075                 4757 ;
R                  2578                 18436                 9259 ;
R                  2578                 15804                13261 ;
```

Figure 2. Standard CIF FORTRAN Output

The first column with “R” just signifies that a “circle” is meant to be generated for each line. Different versions of the generation code have variations on this format, but they are all similar and easily accounted for in the code, and therefore not outlined here.

3.2.2 Microsoft Excel Formatting

The first method developed for converting CIF files to readable files is through manipulating the data manually in Microsoft Excel. First, the data must be imported to an Excel spreadsheet, which can be done by navigating to the “Data” tab in Excel and importing the text file. An example of the code output opened in Microsoft Excel is shown below in Figure 3. Note that when opening the file in Excel, each value should be given its own column. Thus, the “fixed width” and “space delimited” and “tab delimited” options should all be selected when opening the file in Excel. There are two options for readable files to be generated; properly formatted CIF files, or SCR files which can then be converted into DXF files. Both the CIF and DXF file formats can be read by the Heidelberg Instruments DWL 66fs, and both will be described in the following sections.


PS75FL100WL0532HR1.53RANDOMIZED_20151207.CIF - Microsoft Excel
Home
Insert
Page Layout
Formulas
Data
Review
View
Add-Ins
F1
PS75FL100WL0532HR1.53RANDOMIZED
Ready
Average: 0
Count: 4
Sum: 0
100%

	A	B	C	D	E	F	G
1	Photo	n Sieve ring	number =		75		
2	gauss	win alpha =	5	0			
3	focal	length (mic	ron) =	100000	0	0	
4	optim	al wavelengt	h (micron)	0.532	0	0	
5	hole	diameter/rin	g width =	1.53	0	0	
6	comma	nd, d(centim	icron), x(	centimicro	n),	y(centimicron);	
7	R	12476	32619	0	;		
8	R	12476	30136	12482	;		
9	R	12476	23065	23065	;		
10	R	12476	12482	30136	;		
11	R	12476	0	32619	;		
12	R	12476	-12482	30136	;		
13	R	12476	-23065	23065	;		
14	R	12476	-30136	12482	;		
15	R	12476	-32619	0	;		
16	R	12476	-30136	-12482	;		
17	R	12476	-23065	-23065	;		
18	R	12476	-12482	-30136	;		
19	R	12476	0	-32619	;		
20	R	12476	12482	-30136	;		
21	R	12476	23065	-23065	;		
22	R	12476	30136	-12482	;		
23	R	8822	46129	263	;		
24	R	8822	44914	10521	;		
25	R	8822	41446	20252	;		
26	R	8822	35901	28967	;		
27	R	8822	28555	36230	;		
28	R	8822	19777	41675	;		
29	R	8822	10007	45031	;		
30	R	8822	-263	46129	;		
31	R	8822	-10521	44914	;		
32	R	8822	-20252	41446	;		
33	R	8822	-28967	35901	;		
34	R	8822	-36230	28555	;		
35	R	8822	-41675	19777	;		
36	R	8822	-45031	10007	;		
37	R	8822	-46129	-263	;		
38	R	8822	-44914	-10521	;		

Figure 3. FORTRAN Output Opened in Microsoft Excel

3.2.3 Altering Code Output to Match CIF File Syntax

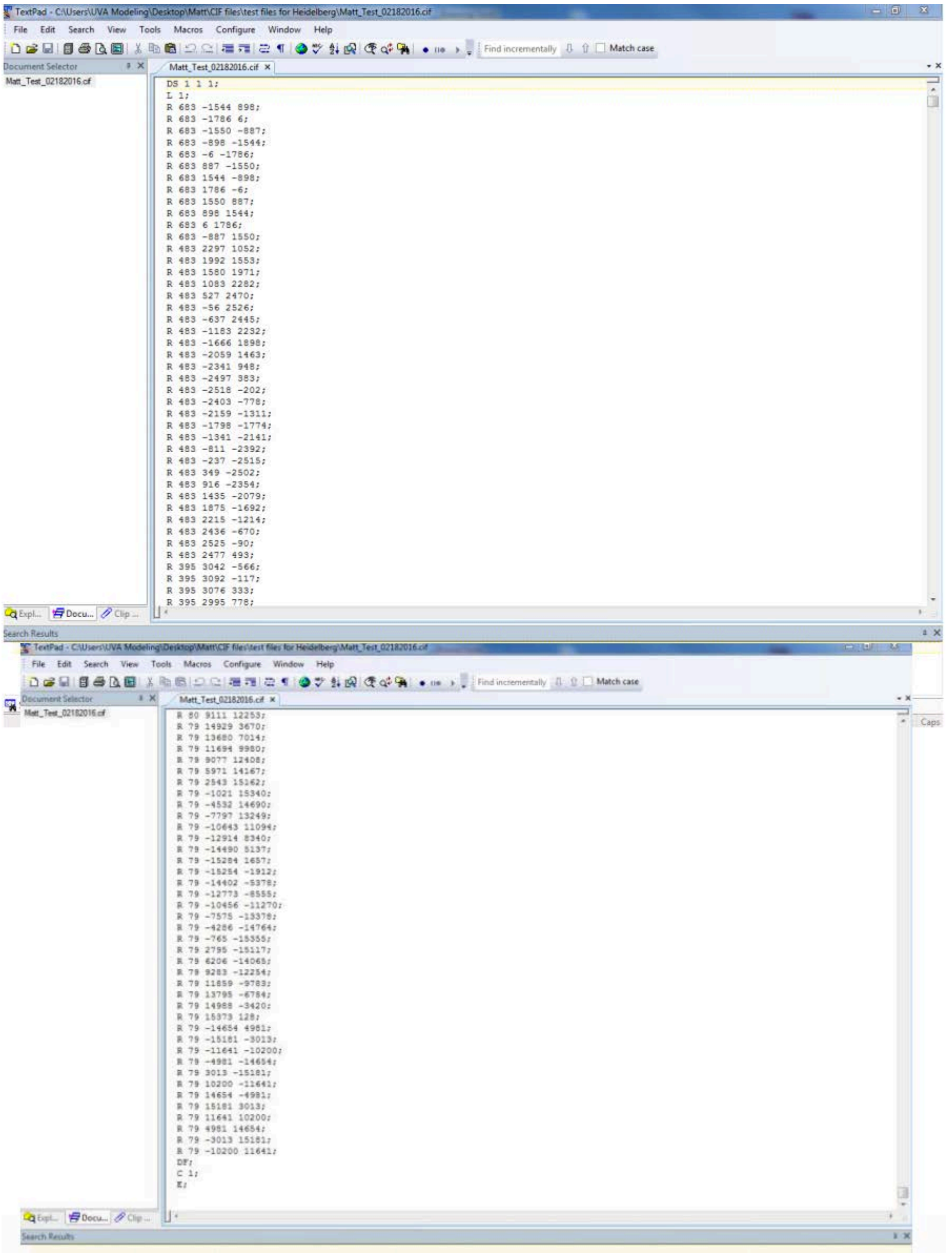
After the data has been exported, it can be edited in Microsoft Excel and TextPad (or any other text editor that can handle large files and perform a search and replace function) to create a usable CIF file. The code output will be manipulated using the following steps:

1) Delete the text header and all other information that is not the hole diameters, x-coordinates, y-coordinates, or semicolons
2) Insert a column before the diameter values and fill the column with “R”

3) Insert a column between each of the columns containing the “R”, diameter, x and y values (the file should now look like: column 1 – R, column 2 – diameter, column 3 – blank, column 4 – x-coordinate, column 5 – blank, column 5, y-coordinate, column 7 – blank, column 8– semicolon)
    a. Four blank columns in total
4) Fill the first three blank columns with a dummy symbol such as the letter “s”
5) Fill the laser blank column with a different dummy symbol such as the letter “p”
    a. These dummies are necessary for the search/replace function
    b. Make sure the dummy symbols run down the entire length of the column
6) Copy all of these columns and paste them into TextPad
    a. Each column should now be separated by a tab in the text editor
7) Perform a search and replace function
8) Search “\ts\t” and replace with “ “ (a single space)
    a. “s” represents the first dummy symbol used
    b. This replaces the “s” columns separated by tabs with a single space, and removes the “s”
9) Search “\tp\t” and replace with “” (no spaces)
    a. “t” represents the second dummy symbol used
    b. This should bring the semicolon directly behind the y-coordinate value with no spaces between them
10) Scroll to the top of the TextPad file and enter the CIF commands for DS and L
11) Scroll to the bottom of the TextPad file and enter the CIF commands for DF, C, and E

Note – be sure that there are no spaces before the “R” at the start of each line. If there is, do a search and replace to replace “ R” with “R”

A finished example of photon sieve parameters successfully converted into CIF file format is shown below in Figure 4.


TextPad - C:\Users\UVA Modeling\Desktop\Matt\CIF files\test files for Heidelberg\Matt_Test_02182016.cif
File Edit Search View Tools Macros Configure Window Help
Find incrementally
Match case
Document Selector
Matt_Test_02182016.cif
Expl... Docu... Clip ...
Search Results
Caps

```
DS 1 1 1;
L 1;
R 683 -1544 898;
R 683 -1786 6;
R 683 -1550 -887;
R 683 -898 -1544;
R 683 -6 -1786;
R 683 887 -1550;
R 683 1544 -898;
R 683 1786 -6;
R 683 1550 887;
R 683 898 1544;
R 683 6 1786;
R 683 -887 1550;
R 483 2297 1052;
R 483 1992 1553;
R 483 1580 1971;
R 483 1083 2282;
R 483 527 2470;
R 483 -56 2526;
R 483 -637 2445;
R 483 -1183 2232;
R 483 -1666 1898;
R 483 -2059 1463;
R 483 -2341 948;
R 483 -2497 383;
R 483 -2518 -202;
R 483 -2403 -778;
R 483 -2159 -1311;
R 483 -1798 -1774;
R 483 -1341 -2141;
R 483 -811 -2392;
R 483 -237 -2515;
R 483 349 -2502;
R 483 916 -2354;
R 483 1435 -2079;
R 483 1875 -1692;
R 483 2215 -1214;
R 483 2436 -670;
R 483 2525 -90;
R 483 2477 493;
R 395 3042 -566;
R 395 3092 -117;
R 395 3076 333;
R 395 2995 778;
```

```
R 80 9111 12253;
R 79 14929 3670;
R 79 13680 7014;
R 79 11694 9980;
R 79 9077 12408;
R 79 5971 14167;
R 79 2543 15162;
R 79 -1021 15340;
R 79 -4532 14690;
R 79 -7797 13249;
R 79 -10643 11094;
R 79 -12914 8340;
R 79 -14490 5137;
R 79 -15284 1657;
R 79 -15254 -1912;
R 79 -14402 -5378;
R 79 -12773 -8555;
R 79 -10456 -11270;
R 79 -7575 -13378;
R 79 -4286 -14764;
R 79 -765 -15355;
R 79 2795 -15117;
R 79 6206 -14065;
R 79 9283 -12254;
R 79 11859 -9783;
R 79 13795 -6784;
R 79 14988 -3420;
R 79 15373 128;
R 79 -14654 4981;
R 79 -15181 -3013;
R 79 -11641 -10200;
R 79 -4981 -14654;
R 79 3013 -15181;
R 79 10200 -11641;
R 79 14654 -4981;
R 79 15181 3013;
R 79 11641 10200;
R 79 4981 14654;
R 79 -3013 15181;
R 79 -10200 11641;
DF;
C 1;
E;
```

Figure 4. Photon Sieve pattern code in CIF Format

Note that in Figure 4, the beginning and end of the code are shown in separate windows, as a typical file contains several thousand or more lines.

Depending on the size of the CIF file, this manual process took hours to complete, as the circles needed to be generated using FORTRAN, then imported and formatted. In addition, the process had to be repeated every time a new sieve was designed, which resulted in an inefficient production cycle. Thus, it became necessary to directly generate the sieves in the correct CIF format.

5.3 Direct CIF Generation

To generate the CIF files directly, the FORTRAN code was implemented into multiple MATLAB programs. MATLAB was chosen as it is designed for large data manipulation, and allows the user to easily write information to text-based files such as CIF files.

5.3.1 CIFgen

CIFgen is the first and most straightforward MATLAB generator written. Its inputs are as follows:

1) Ring Count
2) Focal Length (mm)
3) Wavelength (nm)
4) D/W ratio – ratio of hole diameter to Fresnel zone plate gap diameter
5) Gauss alpha – Determines hole spacing, higher values yield denser holes
6) Gauss beta – Determines hole spacing, higher values yield denser holes

Any of these values can be input as 0, which will input a default, standard value hard coded in. The program produces a CIF file in the current directory with the specified properties. The file name consists of these properties, which allows for easily distinguishing various sieve designs. The program also writes certain statistics to the command window, including hole count, smallest hole size, and sieve size. These values are labeled with units. The program as a whole, along with a description of the function, can be found in Appendix A.

However, CIFgen is limited by the memory of the computer that the program is run on. If the input parameter for the ring count exceeds approximately 10,000 rings, the variables become too large for a computer with 16 GB or less of memory to store. As a result, it became necessary to write a new program that could account for sieves of larger sizes.

5.3.2 FrankensteinCIF

This program, modeled after CIFgen, constructs a sieve in segments to avoid the issue of large variables. The name “Frankenstein” is inspired by Frankenstein’s monster; like the fictional character, the photon sieve designs created by this program comprise of different sections stitched together. FrankensteinCIF contains the same inputs as CIFgen. The program works by constructing the sizes and coordinates of holes for a range of rings, using two helper functions.

The first helper function used is called HoleCounter. This function uses the inputs of FrankensteinCIF to determine the number of rings that can be processed while still remaining below the computer's memory threshold. When the number of holes counted exceeds the amount that the computer's memory can handle, the function then determines the number of rings that can be processed at one time. The outputs of this helper function are used in the other helper function, ExactVectorsCIF, which determines the hole information for the calculated range of rings.

ExactVectorsCIF is written with the following inputs:

1) Maximum Ring Count
2) Focal Length (mm)
3) Wavelength (nm)
4) D/W ratio – ratio of hole diameter to Fresnel zone plate gap diameter
5) Gauss alpha – Determines hole spacing, higher values yield denser holes
6) Gauss beta – Determines hole spacing, higher values yield denser holes
7) Minimum Ring Count

ExactVectorsCIF writes the x-coordinates, y-coordinates, and diameters of the holes to corresponding vectors. These vectors consist of only the information for the range of rings (from the minimum input to the maximum input). After this, HoleCounter determines the next range of rings, and the process repeats. Then, in FrankensteinCIF, these vectors are concatenated; like the namesake of the program (Frankenstein's Monster), the individual parts are stitched together in order to create a larger CIF file. A demonstration file is shown in Figure 5, with intentional gaps between the sections of the sieve. Note that the size of the sections decreases as the sieve expands outwards, due to the smaller size of the holes. This results in more holes per ring, which decreases the amount of rings that can be processed at one time.

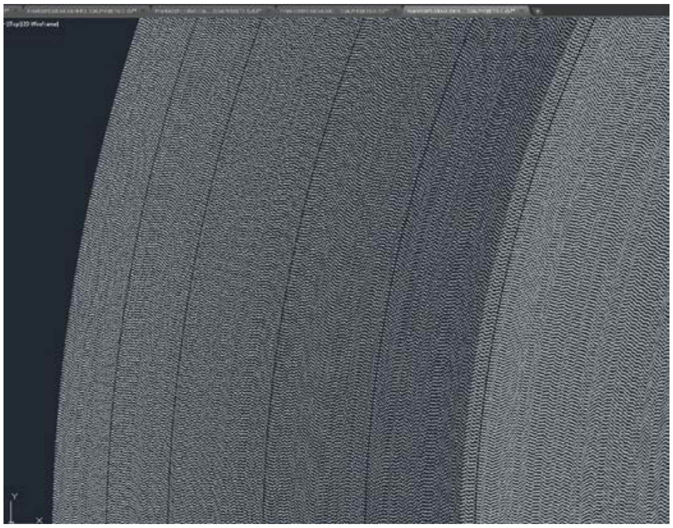


Figure 5. FrankensteinCIF Section Demonstration

By constructing the coordinates in parts, none of the variables become too large for the computer to process. In the case that the desired sieve is small enough for the program to compute in one piece, CIFgen is automatically run instead. Due to the ability to generate sieves of all sizes, FrankensteinCIF is the preferred CIF generation method. FrankensteinCIF, ExactVectorsCIF, and HoleCounter are all located and described in detail in Appendix A.

5.4 Previewing CIF Files

Once the CIF file has been loaded into the Heidelberg DWL 66fs conversion software, it can be previewed for size, errors, and other properties. However, this preview is severely limited by the memory of the PC that the conversion software runs on. It was determined through trial and error using CIF files of different sizes that the preview is limited to sieves with 20,000,000 holes or less; a larger amount of holes causes the computer to freeze due to insufficient memory. Therefore, it became necessary to preview the CIF files prior to manufacturing, using an alternative software, KLayout.

KLayout is an open-source program built for Windows, Macintosh, and Linux users [5]. It allows for the viewing and conversion of many different layout formats, including CIF, DXF, GDSII, and Gerber, all of the file formats compatible with the Heidelberg DWL 66FS. An example preview in KLayout is shown in Figure 6. The sieve shown has a diameter of 6 inches, with a wavelength of 1064 nm and a focal length of 1 m.

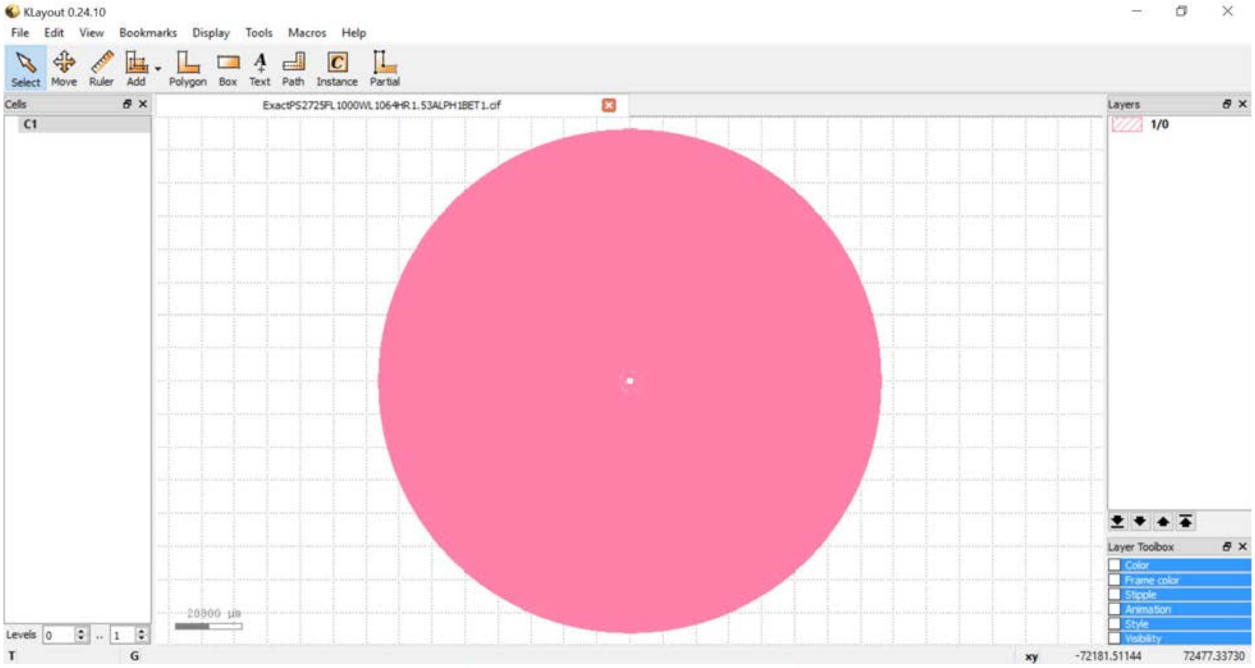


Figure 6. Preview of a 6-inch Diameter Photon Sieve in KLayout

On a computer with sufficient memory, KLayout enables the user to view the CIF file, measure features such as the hole size and spacing, and save as a different format, among other features. The ability to measure features in KLayout is shown in Figure 7. The sieve shown is the same 6-inch diameter sieve as shown in Figure 6, zoomed in. The measurements shown are in microns.

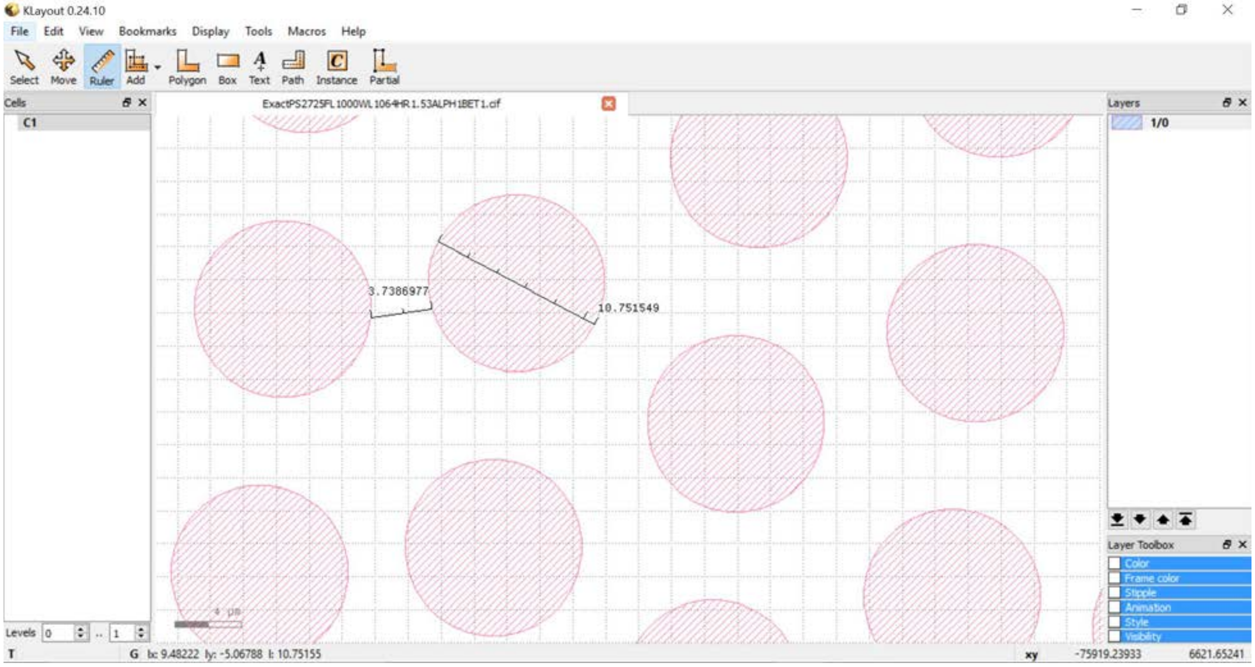


Figure 7. Hole Diameter and Spacing Measurements in KLayout

Using KLayout, the user can ensure that the file meets specifications without having to bring it to the clean room to test, increasing time efficiency.

5.5 CIF Conclusion

When the manual conversion and the direct generation of the CIF files are compared, it is clear that the direct generation of the CIF files is the more efficient and therefore preferred approach. With the use of a direct MATLAB generation program, it becomes simple to quickly produce sieves of varying parameters, without the need to manually format each one. With the successful generation, as well as preview of photon sieves using this process, there is no need to further improve upon the CIF file generation.

6. DXF File Generation

There are various methods to generate DXF files, which are described in the section below. First, the file format is outlined, along with examples of how the file may be written. Next, the initial manual method of generating DXF files using a FORTRAN output and Microsoft Excel is described. Finally, this report details the MATLAB programs written to directly generate DXF files from a set of given parameters.

6.1 DXF File Format

Drawing Interchange Format, or DXF, is a file format developed by Autodesk, Inc. to represent the information contained in an AutoCAD drawing file [6]. This file format is supported in an ASCII form as well as a binary form, with the ASCII form more commonly used. Every DXF file consists of numerical group codes, followed by numbers or phrases containing the relevant information. These group codes, as well as the various sections and syntax of the DXF format are described in the DXF Reference, which Autodesk, Inc. publishes for the convenience of AutoCAD users. The relevant contents of the file are discussed, followed by a description of the header that had to be added to the file.

6.1.1 DXF File Content

The following is a depiction of an example photon sieve DXF file. As seen below, the format is redundant, with a group code describing the information that follows. Each piece of information is written on a new line. The following example shows the DXF file (written in bold) followed by a description of what that line would do in a file (written in parentheses). All numbers are unitless, so the user must specify the base unit used when the design is loaded to the Heidelberg DWL 66fs.

0 (Following line is a String)

SECTION (New Section)

2 (Following line is a Name)

ENTITIES (Name of the Section)

0 (Following line is a String)

CIRCLE (Entity Type)

8 (Following line is the Layer Name)

LAYER1 (Layer Name)

10 (Following line is X Value of the Primary Point)

0.000000 (X Value of the Primary Point)

20 (Following line is Y Value of the Primary Point)

0.000000 (Y Value of the Primary Point)

40 (Following line is the radius of the circle)

0.010000 (Radius of the circle)

0 (Following line is a String)

CIRCLE (Entity Type)

8 (Following line is the Layer Name)

LAYER1 (Layer Name)

10 (Following line is X Value of the Primary Point)

200.000000 (X Value of the Primary Point)

20 (Following line is Y Value of the Primary Point)

200.000000 (Y Value of the Primary Point)

40 (Following line is the radius of the circle)

0.500000 (Radius of the circle)

0 (Following line is a string)

ENDSEC (End of section)

0 (Following line is a string)

EOF (End of File)

For the generation of a photon sieve, the circle is the only relevant entity type. Every new circle receives a new 10 lines of information describing that circle. Each design must follow the format described above, as well as the Heidelberg Instruments common design rules shown earlier (see 3. Heidelberg DWL 66fs System). In addition, there are specific rules for the DXF format, described below.

- A 100% AutoCAD compatible editor must be used
- All polylines must be joined properly
    - Be especially cautious when using arcs within a polyline.
- Only one font is provided with the DXF conversion package. Any other font will be replaced by this standard font.
- Only the following attributes assigned to a text are supported: ROTATION, MIRRORING, SCALING.
- Avoid putting structures to layer 0.
    - The MATLAB program directly writes the structures to layer “1” to avoid this problem (See Appendix B)
    - If the design is generated manually, the layer must be changed using the AutoCAD Layer Properties Manager (command “AMLAYMOVE”)
- Do not use special characters in the layer names.

- Try to design in the metric system! Use mm as the standard unit. (Some design programs will not accept inches!
- Polylines with widths must not have a change in its widths (Tapered lines).
- Different scaling in x and y when inserting a block is not supported.
- External blocks are not supported.
- Avoid using BLOCK insertions. Try to make the design flat.
  - Use the conversion software step and repeat function instead (keeps the DXF data file small!).
- Try to use only the following entities: CIRCLE, POLYLINE (but see the remark above), and TEXT.

Besides the file content described above, the only information in a DXF file is the header, which is the most complicated component.

6.1.2 DXF File Header

Due to the amount of commands present in the DXF format, alternative means were used to determine the content of the file header. A DXF file consisting solely of the information described above was loaded to a conversion software called LinkCAD, which had the ability to read and convert multiple different layout formats. When the original, user-generated DXF file was converted to a LinkCAD DXF file (using LinkCAD's built in DXF file format), 277 lines were added to the beginning of the file. Out of these 277 lines, only 8 of them differed from sieve to sieve; these files were omitted with no impact to the file outcome. Therefore, the remaining 269 lines were added as a header. These lines are shown in Appendix B as a part of the MATLAB generation code (to be explained at a later point).

The DXF file must follow the format shown above exactly in order to ensure successful processing by the Heidelberg DWL 66fs. If the format is incorrect, the instrument will display an error when attempting to read the file.

6.2 Manual DXF Conversion

To create a DXF file (AutoCAD file) from the FORTRAN output described earlier, a different set of steps was followed. To generate a DXF file, the code output must be converted into an AutoCAD script (SCR file), the script must be opened in AutoCAD, and then finally saved as a DXF file.

6.2.1 Syntax for SCR Files

Similar to CIF files, SCR files have their own syntax that must be present in order for the file to be successfully read by the Heidelberg DWL66fs and by AutoCAD. There are no extra commands other than the geometric parameters for SCR files, so the circle information must simply be transformed into the correct format. The format is as follows:

circle x,y radius

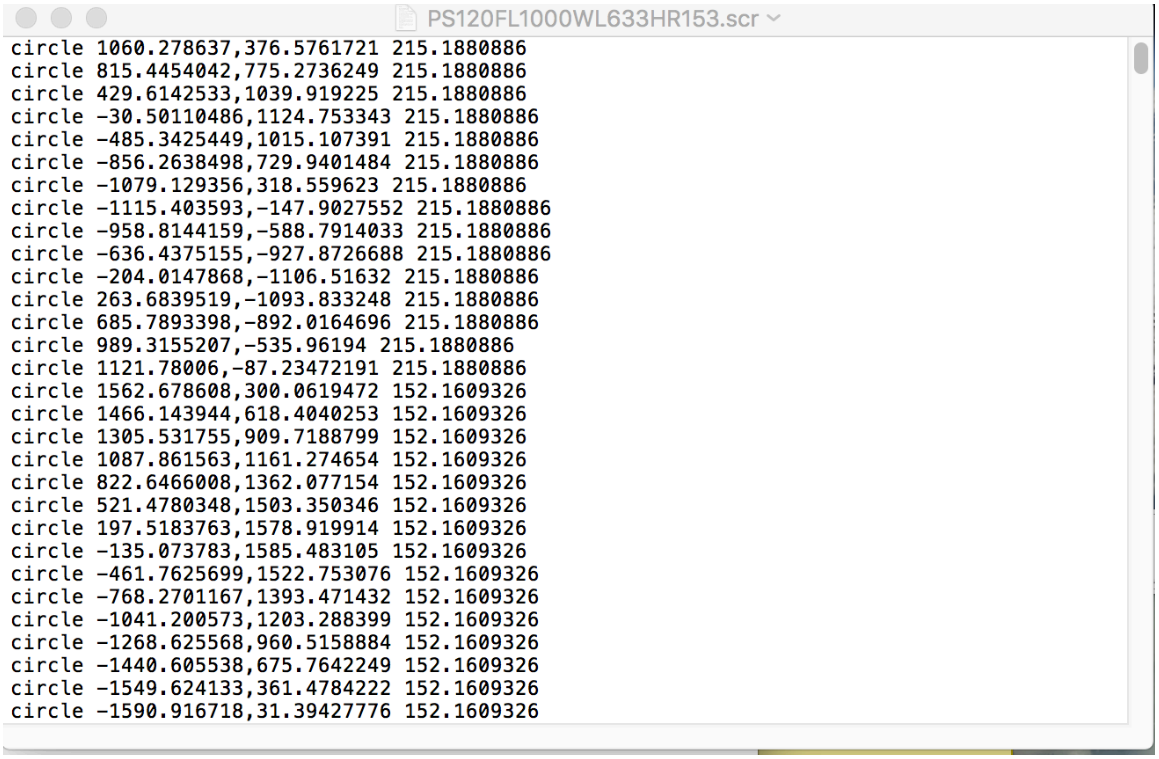
Where x, y, and radius are replaced with the corresponding values for the circles. For the purposes of photon sieves, the units for all quantities are in microns. There must be no blank lines in the SCR file. Each line must contain a command.

6.2.2 Generating the SCR File

In order to generate the SCR file from the code, the FORTRAN output must be manipulated, similar to the CIF conversion. The steps are similar, and they are as follows (note that numerical values need not be integers):

1. The Fortran output is opened in Excel
2. The text header and all other information that is not the hole diameters and x, y-coordinates is deleted
3. The columns are reordered so that, from left to right, they are in the order of x coordinate, y coordinate, diameter
4. The diameter column is divided by 2 to obtain the corresponding radius
5. All columns are multiplied by 100 to convert the units from centimicrons to microns
6. A column is inserted before the x values. Fill the length of the column with the word “circle”
7. Dummy columns are inserted between all four columns
8. The columns following “circle” and the y coordinates are filled with “s” (arbitrary dummy symbol 1)
9. The column following the x coordinate is filled with “p” (arbitrary dummy symbol 1)
10. Columns should now be in the order (from left to right): circle, s, x coordinate, p, y coordinate, s, radius
11. All columns are copied and pasted into TextPad (or other text editor with search and replace function)
12. A search and replace is performed
    a. “\ts\t” is searched and replaced with “ “ (a single space)
    b. “\tp\t” is searched and replaced with “,” (a comma with no spaces)
13. The file is saved, adding “.scr” to the end of the file name
    a. For “save as type” select “All Files (*.*)”
    b. For example, the file name may be “example.scr”
    c. This will convert the .txt file to a SCR file

Note that the symbols “s” and “p” are arbitrary symbols and can be replaced with any other symbols, as long as consistency throughout the conversion process is retained. An example of a finished SCR file is shown in Figure 8.



```
circle 1060.278637,376.5761721 215.1880886
circle 815.4454042,775.2736249 215.1880886
circle 429.6142533,1039.919225 215.1880886
circle -30.50110486,1124.753343 215.1880886
circle -485.3425449,1015.107391 215.1880886
circle -856.2638498,729.9401484 215.1880886
circle -1079.129356,318.559623 215.1880886
circle -1115.403593,-147.9027552 215.1880886
circle -958.8144159,-588.7914033 215.1880886
circle -636.4375155,-927.8726688 215.1880886
circle -204.0147868,-1106.51632 215.1880886
circle 263.6839519,-1093.833248 215.1880886
circle 685.7893398,-892.0164696 215.1880886
circle 989.3155207,-535.96194 215.1880886
circle 1121.78006,-87.23472191 215.1880886
circle 1562.678608,300.0619472 152.1609326
circle 1466.143944,618.4040253 152.1609326
circle 1305.531755,909.7188799 152.1609326
circle 1087.861563,1161.274654 152.1609326
circle 822.6466008,1362.077154 152.1609326
circle 521.4780348,1503.350346 152.1609326
circle 197.5183763,1578.919914 152.1609326
circle -135.073783,1585.483105 152.1609326
circle -461.7625699,1522.753076 152.1609326
circle -768.2701167,1393.471432 152.1609326
circle -1041.200573,1203.288399 152.1609326
circle -1268.625568,960.5158884 152.1609326
circle -1440.605538,675.7642249 152.1609326
circle -1549.624133,361.4784222 152.1609326
circle -1590.916718,31.39427776 152.1609326
```

Figure 8. Example SCR File

6.2.3 Saving the SCR File as a DXF File

Once the script file has been successfully created, it can be opened in AutoCAD. To do so, AutoCAD must be opened, and “script” must be typed into the command bar. When the enter key is pressed, the software should open a browse window where the user can locate the script file. Once the script file is opened, the photon sieve pattern should appear in AutoCAD. To change the AutoCAD units to microns, “units” is typed into the command bar the enter key is pressed. “Microns” will be pressed from the drop down menu for units, and after the “OK” key is pressed, the units should be changed successfully.

Finally, the document can be saved as a “DXF” file using the “Save As” button. An image of a photon sieve in AutoCAD is shown below in Figure 9.

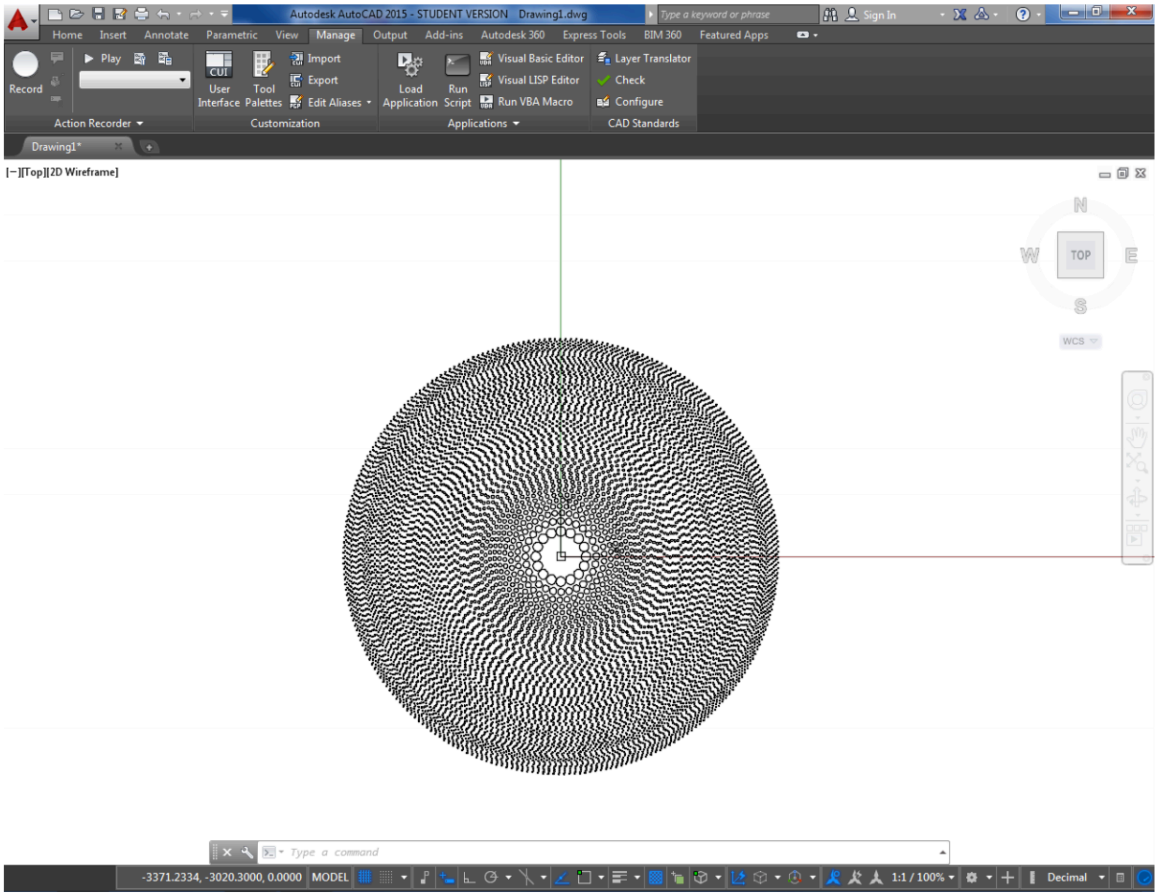


Figure 9. A photon sieve opened in AutoCAD after SCR conversion

Similar to the CIF file conversion, this process is tedious and time-consuming. Due to the inefficiency, it became necessary to create a direct means of generating DXF files in the proper format.

6.3 Direct DXF Generation

To generate the DXF files directly, the FORTRAN code was implemented into multiple MATLAB programs. MATLAB was chosen as it is designed for large data manipulation, and allows the user to easily write information to text-based files such as DXF files. The programs described are identical in function to the CIF conversion programs described in section 5.3, and differ only in the file writing syntax.

6.3.1 ExactGen

ExactGen is the first and most straightforward MATLAB generator written. Its inputs are as follows:

1) Ring Count
2) Focal Length (mm)
3) Wavelength (nm)
4) D/W ratio – ratio of hole diameter to Fresnel zone plate gap diameter
5) Gauss alpha – Determines hole spacing, higher values yield denser holes
6) Gauss beta – Determines hole spacing, higher values yield denser holes

Any of these values can be input as 0, which will input a default, standard value hard coded in. The program produces a DXF file in the current directory with the specified properties. The file name consists of these properties, which allows for the easy distinguishing of various sieves. The program also writes certain statistics to the command window, including hole count, smallest hole size, and sieve size. These values are labeled with units. The program and description of the function can be found in Appendix B.

However, ExactGen is limited by the memory of the computer that the program is run on. If the input parameter for the ring count exceeds approximately 10,000 rings, the variables become too large for a computer with 16 GB or less of memory to store. As a result, it became necessary to write a new program that could account for sieves of larger sizes.

6.3.2 FrankensteinMonster

This program, modeled after ExactGen, constructs a sieve in segments to avoid the issue of large variables. FrankensteinMonster contains the same inputs as ExactGen. The program works by constructing the sizes and coordinates of holes for a range of rings, using two helper functions.

The first helper function used is HoleCounter (see 5.3 Direct CIF Generation). The outputs of this helper function are used in the other helper function, ExactVectors, which determines the hole information for the calculated range of rings.

ExactVectors is written with the following inputs:

1) Maximum Ring Count
2) Focal Length (mm)
3) Wavelength (nm)
4) D/W ratio – ratio of hole diameter to Fresnel zone plate gap diameter
5) Gauss alpha – Gauss values determine hole spacing, higher values yield denser holes
6) Gauss beta

7) Minimum Ring Count

ExactVectors writes the x-coordinates, y-coordinates, and diameters of the holes to corresponding vectors. These vectors consist of only the information for the range of rings (from the minimum input to the maximum input). After this, HoleCounter determines the next range of rings, and the process repeats. Then, in FrankensteinMonster, these vectors are concatenated. The individual parts are stitched together in order to create a larger DXF file (like the namesake of the program - Frankenstein's Monster). A demonstration file is shown in Figure 10, with intentional gaps between the sections of the sieve. Note that the size of the sections decreases as the sieve expands outwards, due to the smaller size of the holes. This results in more holes per ring, which decreases the amount of rings that can be processed at one time.

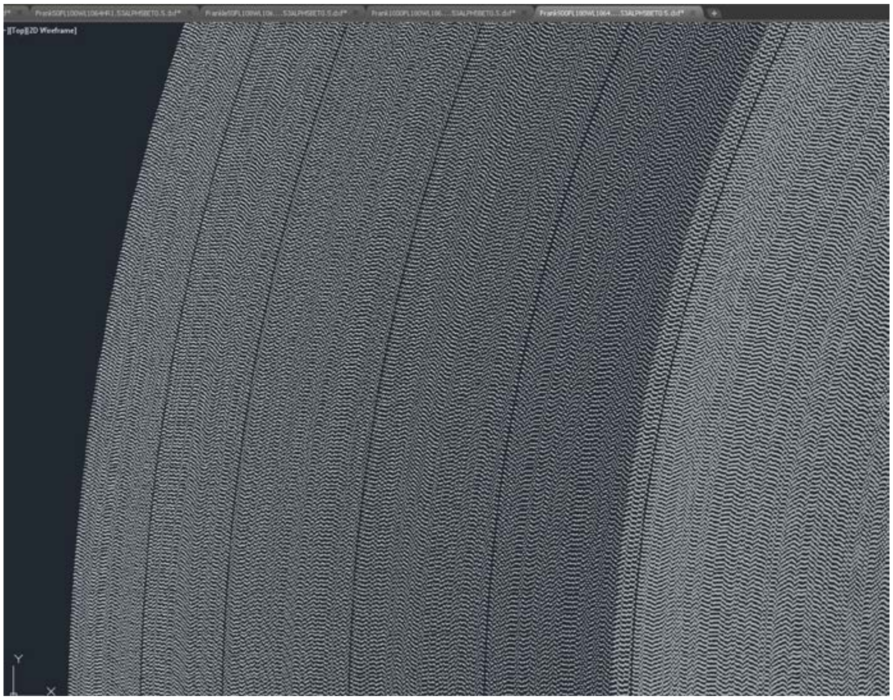


Figure 10. FrankensteinMonster Section Demonstration

By constructing the coordinates in parts, none of the variables become too large for the computer to process. In the case that the desired sieve is small enough for the program to compute in one piece, ExactGen is automatically run instead. Due to the ability to generate sieves of all sizes, FrankensteinMonster is the preferred DXF generation method. FrankensteinMonster and ExactVectors are located and described in detail in Appendix B.

6.3.3 ExactGen_Parts

This program allows the user to generate a photon sieve design as separate DXF files, which can be combined when manufacturing the sieve. It functions in the same way as ExactGen, except that the user can input the number of files that the file needs to be split into. The function then splits the hole information evenly into that number of parts, and writes this information to separate files.

Information for each file is displayed in the Command Window, such as the smallest and largest hole sizes, and the diameter of that individual part. ExactGen_Parts is located and described in detail in Appendix B.

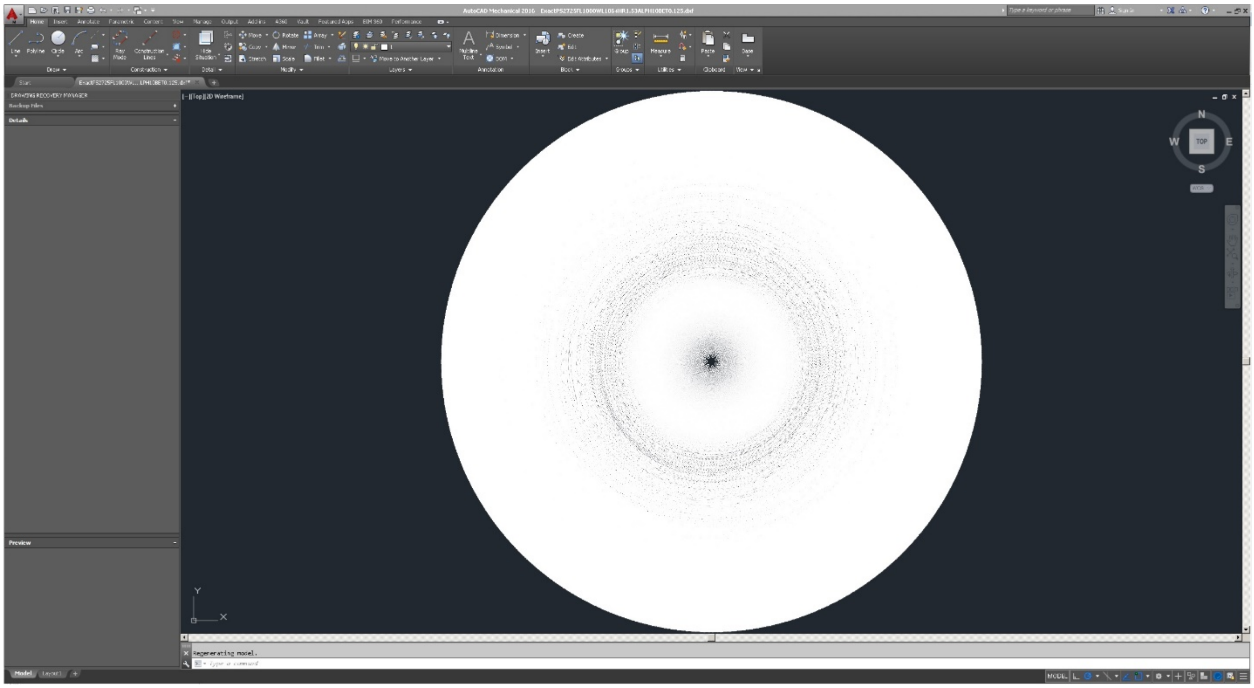
6.3.4 PhotonSieve_Generation

This program allows the user to create a sieve of either file type, using the same inputs as the previous programs as well as an additional variable for the file format. Depending on the user inputs, the program runs one of the programs previously described. This program is the most useful, as the other programs written are all nested within "PhotonSieve_Generation." Due to this, this file is stand-alone, and can be transported easily between computers. The function is located and described in detail in Appendix B.

6.4 Previewing DXF Files

Like CIF files, DXF files can be conveniently previewed by the Heidelberg DWL66fs conversion software to check for errors, hole size, and hole spacing. Similar to the CIF files (See 5.4 Previewing CIF Files), however, the memory of the computer limits the ability to preview large DXF files. It was determined through trial and error that the conversion software was only able to preview files with less than approximately 20,000,000 holes. One alternative to view the DXF files is KLayout, the open-source software also used to preview CIF files. However, DXF files have the advantage of being compatible with AutoCAD, a common layout and design software. AutoCAD is a software familiar to many scientists and engineers, which eliminates the learning curve of having to use a new software such as KLayout. This increases the efficiency and appeal of DXF files. However, AutoCAD is a more memory intensive application than KLayout, so on a computer with little memory, previewing large sieve designs may cause slow performance.

An example of a DXF file viewed AutoCAD is shown below.



```
Regenerating model.
```

Figure 11. Photon Sieve in AutoCAD

As with KLayout, the user can measure features such as hole size and spacing in AutoCAD. With the combination of the two preview software programs, the DXF files can easily be checked prior to manufacturing.

6.5 DXF Conclusion

When the manual conversion and the direct generation of the DXF files are compared, it is clear that the direct generation of the DXF files is the more efficient and therefore the preferred approach. However, with the complicated nature of the DXF format, it is difficult to understand the full process of adding a header and using the correct syntax. With the use of a direct MATLAB generation program, it became simple to quickly produce sieves of varying parameters, without the need to manually format each one. With the successful generation as well as preview of photon sieves using this process, there is no need to further improve upon the DXF file generation.

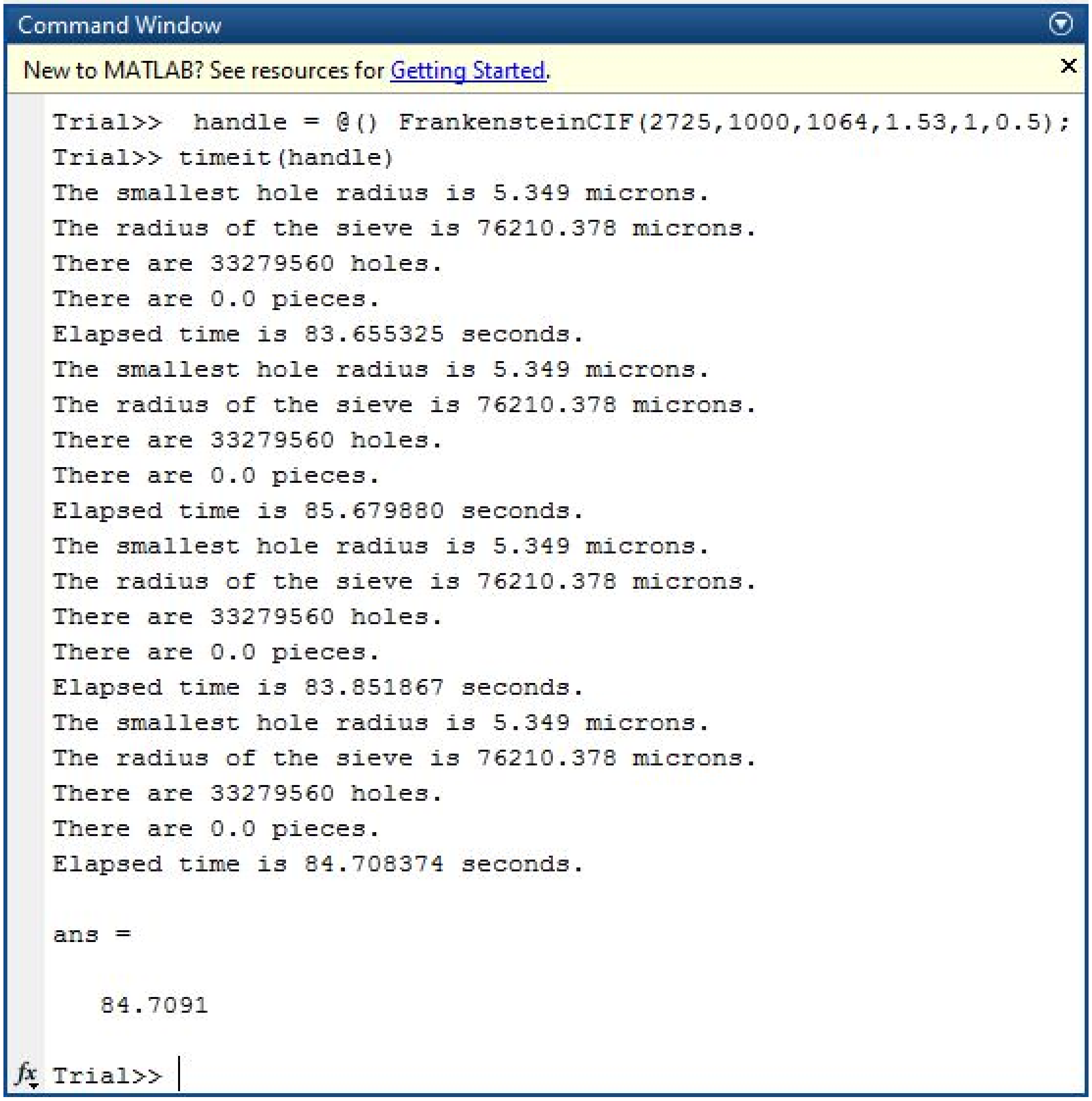
7. CIF and DXF Comparison

Due to the ability to generate photon sieves in two different file formats, it became necessary to compare the CIF and DXF formats and analyze them to determine a preferable approach. The accuracy, size, complexity, computation time, and available resources for each file will be considered. The results for each of these categories will be presented in the form of a decision matrix, which will offer an objective comparison of the strengths and weaknesses of each format. Table 1 displays information concerning sieves produced using each of the file format methods. In both cases, the sieve consisted of the same input parameters; the only variable was the format the file was written in. Differences between the formats are highlighted for convenience.

Table 1. CIF and DXF Production Results

	CIF File	**DXF File**
Sieve Radius (microns)	76,210.378	76,210.378
Number of Rings	2725	2725
Number of Holes	33,279,560	33,279,560
Smallest Hole Radius (microns)	5.349	5.349
Computing Time (seconds)	84.7091	93.7212
Focal Length (mm)	1000	1000
Wavelength (nm)	1064	1064
D/W Ratio	1.53	1.53
Gauss Alpha	1	1
Gauss Beta	0.5	0.5
File Size	880 MB	1.51 GB

Values such as the sieve radius, number of holes, and smallest hole radius were displayed in the MATLAB command window after the file had been produced. The specific functions used in to display this information are located in the MATLAB code repository, found in Appendix A and Appendix B. To determine the computing time, the MATLAB function "timeit" was used. This function runs each program multiple times to determine the average runtime. The command window displays that show the corresponding information in Table 1 are located below.



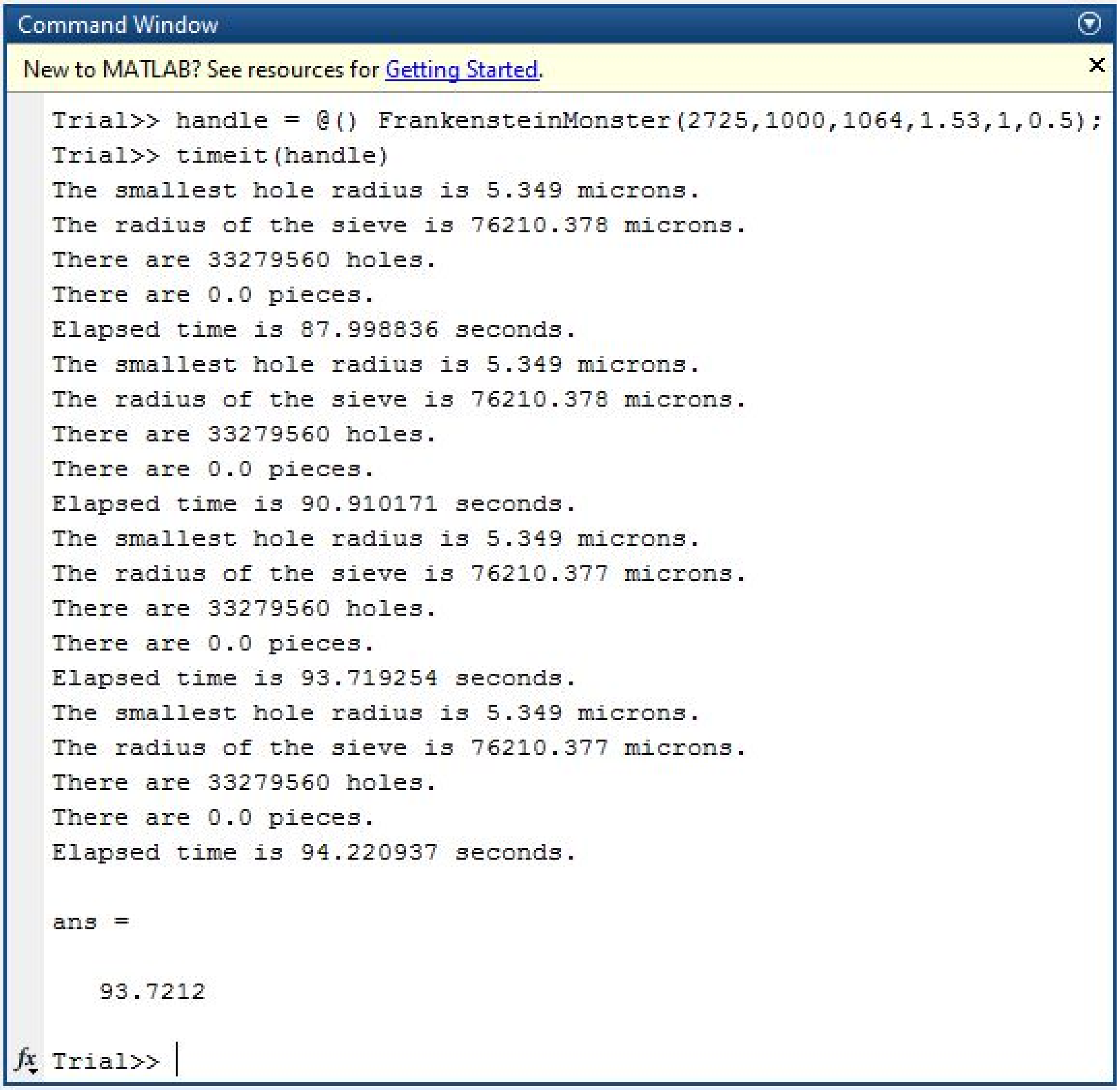
```
Trial>>  handle = @() FrankensteinCIF(2725,1000,1064,1.53,1,0.5);
Trial>> timeit(handle)
The smallest hole radius is 5.349 microns.
The radius of the sieve is 76210.378 microns.
There are 33279560 holes.
There are 0.0 pieces.
Elapsed time is 83.655325 seconds.
The smallest hole radius is 5.349 microns.
The radius of the sieve is 76210.378 microns.
There are 33279560 holes.
There are 0.0 pieces.
Elapsed time is 85.679880 seconds.
The smallest hole radius is 5.349 microns.
The radius of the sieve is 76210.378 microns.
There are 33279560 holes.
There are 0.0 pieces.
Elapsed time is 83.851867 seconds.
The smallest hole radius is 5.349 microns.
The radius of the sieve is 76210.378 microns.
There are 33279560 holes.
There are 0.0 pieces.
Elapsed time is 84.708374 seconds.

ans =

   84.7091

fx Trial>>
```

Figure 12.   MATLAB Command Window with CIF Photon Sieve Data



```
Trial>> handle = @() FrankensteinMonster(2725,1000,1064,1.53,1,0.5);
Trial>> timeit(handle)
The smallest hole radius is 5.349 microns.
The radius of the sieve is 76210.378 microns.
There are 33279560 holes.
There are 0.0 pieces.
Elapsed time is 87.998836 seconds.
The smallest hole radius is 5.349 microns.
The radius of the sieve is 76210.378 microns.
There are 33279560 holes.
There are 0.0 pieces.
Elapsed time is 90.910171 seconds.
The smallest hole radius is 5.349 microns.
The radius of the sieve is 76210.377 microns.
There are 33279560 holes.
There are 0.0 pieces.
Elapsed time is 93.719254 seconds.
The smallest hole radius is 5.349 microns.
The radius of the sieve is 76210.377 microns.
There are 33279560 holes.
There are 0.0 pieces.
Elapsed time is 94.220937 seconds.

ans =

   93.7212

Trial>> 
```

Figure 13. MATLAB Command Window with DXF Photon Sieve Data

7.1 File Accuracy

As seen in Table 1, both the CIF file and DXF file produced sieves of the exact same physical parameters. Both sieves had radii of 3 inches, with the same number of holes. Although neither sieve has been produced to physically test optical accuracy, the sieves are physically identical, and would therefore hold the same margin of error. Due to this, it can be stated that sieves produced using the CIF method and the

DXF method are identical in terms of accuracy. This is an expected result, as the functions that produced both sieves have the same core model.

7.2 File Size

One of the largest differences between the CIF and DXF file formats is the resulting file size. As seen in Table 1, a six inch diameter sieve with the given parameters resulted in an 880 MB CIF file, and a 1.51 GB DXF file. The size ratio of approximately 1.72:1 for DXF: CIF is not catastrophic in this scenario, where both files are relatively small. However, for sieves with smaller focal lengths, and therefore many more rings and holes, the file size dramatically affects the convenience and performance. For example, a sieve with a focal length of 100 cm and 24176 rings (six inch diameter) is 55.5 GB in the DXF format, and 30.4 GB in the CIF format. This ratio of approximately 1.83:1 affects the processing time and convenience in this situation. With such large files, factors such as the transportation to the clean room, the machine reading time, and even desktop storage become increasingly complicated. In scenarios such as those, it becomes necessary to keep the file size as small as possible, which a CIF file accomplishes. Due to this, CIF files hold a strong advantage over DXF files in terms of size and storage.

7.3 Format Complexity

As discussed earlier in the report (see section 5.1), the CIF file format is simple, with only six relevant commands for photon sieve generation. This allows the user to easily understand and manipulate the information in the file, even after generation. The DXF format (see section 6.1), on the other hand, is both redundant and complex, which limits the ability of the user to understand and modify the contents of the file. Due to the creation of the direct generation programs (see sections 5.3 and 4.3), it became possible to create the both the CIF and DXF files without understanding the specific format syntax. This made it unnecessary to completely know the syntax; however, the CIF still holds the advantage over the DXF as it allows the user to easily have the editing option if desired.

7.4 Computation Time

As shown in Table 1, there was a slight difference of 9.0121 seconds between the computation times of the DXF file and the CIF file. The generation of both files was completed on a Dell Precision T3600 PC, with an Intel Xeon E5-1650 processer (base frequency of 3.20 GHz) and 32 GB of installed memory (RAM). The specific computation time depends on the computer used; however, the time difference will remain similar due to the difference in the file formats. The DXF file is slightly slower, in part due to the need to write every new piece of information to a different line. The CIF file, on the other hand, contains all of the information for one circle on one line, which results in a faster program. This difference remains small even when the sieves are increased in size (within 5 minutes on a production time of approximately 1 hour). Due to the small size of the time difference, there is little to no impact on the overall sieve production efficiency. Because of this, the slight advantage of the CIF file holds little weight.

7.5 Available Resources

Both the DXF and CIF formats have readily available documentation describing the use and syntax of the file formats. However, the DXF syntax changes as new versions of AutoCAD are released, which makes it difficult to obtain the correct documentation and retain the same format throughout a large file. The CIF format, on the other hand, has remained constant since 1980; this consistency ensures that there are no differences in files produced at different times.

Although the DXF format changes occasionally, the documentation as a whole is by far more thorough than the CIF documentation, with a step-by-step description of every possible command, its use, and examples of code [6]. The CIF format reference, on the other hand, is a brief document that verbally explains how to use the format rather than visually explaining through examples [4]. Due to this, it is therefore necessary to find external CIF examples to fully understand the format, which is difficult due to the age of the format. Therefore, although the DXF format changes, the thorough documentation results in a strong advantage over the CIF format.

7.6 Decision Matrix

The following matrix contains the results of the analysis described above. Each category is given a weight (which sum to 1), and the DXF and CIF file are scored from -1 to 1 for each category. A score of -1 represents a strong disadvantage, a score of 0 represents neither an advantage nor a disadvantage, and a score of 1 represents a strong advantage.

Table 2. File Format Decision Matrix

Category	Weight	CIF File	DXF File
File Accuracy	0.3	0	0
File Size	0.3	1	-1
Format Complexity	0.1	1	-1
Computation Time	0.2	0.5	0
Available Resources	0.1	-0.5	0.5
Total	1.0	0.45	-0.35

As shown by the decision matrix, the CIF file is preferable to the DXF file, particularly to the ease of use and smaller file size.

8. Conclusion

This report has detailed various methods of producing photon sieve design files using both manual and automated conversion techniques. The two file types used, DXF and CIF, are both compatible with the Heidelberg DWL66fs laser lithography system, and are compatible with various commercial preview and editing programs. When the two file formats are compared, CIF files are clearly more user-friendly and efficient in storing the information, which makes this format the preferred file type for photon sieve design generation. With the completion efficient and simple design generation process, photon sieves can now be manufactured at a higher rate for imaging applications.

9. References

[1] L. Kipp, M. Skibowski, R. L. Johnson, R. Berndt, R. Adelung, S. Harm and R. Seemann, "Sharper images by focusing soft X-rays with photon sieves," Nature, 2001.

[2] G. Andersen, M. E. Dearborn, M. G. McHarg and J. Harvey, "Membrane Photon Sieve Telescope," SPIE 8385, Sensors and Systems for Space Applications V, Baltimore, Maryland, 2012.

[3] Heidelberg Instruments, "DWL66fs User Guide," Heidelberg Instruments, Heidelberg, Germany, 2015.

[4] R. Sproull, R. Lyon and S. Trimberger, "The Caltech Intermediate Form for LSI Layout Description," California Institute of Technology, 1980.

[5] "KLayout - High Performance Layout Viewer And Editor," [Online]. Available: http://www.klayout.de/index.html. [Accessed 29 June 2017].

[6] Autodesk, Inc., "DXF Reference," Autodesk, Inc, San Rafael, CA, 2011.

Appendix A: CIF Code Repository

NOTE: All the code can be provided by request.

1. CIFgen

CIFgen, the first photon sieve conversion program written, reads an input of photon sieve design parameters, and directly outputs a CIF file with the design. This program also displays information about the sieve, such as the radius, size of the smallest hole, and number of holes.

1.1 CIFgen Full Function

2. ExactVectorsCIF

ExactVectorsCIF takes a range of rings and writes the information for each circle to vectors. The function then repeats the process for the following range of rings until it reaches the end of the sieve.

3. HoleCounter

HoleCounter takes the range of rings (minimum and maximum) as well as the other photon sieve design parameters to count the number of holes in each section of the sieve. This is used in FrankensteinCIF to separate the sieve into sections that fit within the computer's memory.

3.1 HoleCounter Full Function

4. FrankensteinCIF

This function uses ExactVectorsCIF and HoleCounter to split the sieve into smaller sections. The hole information for each section is calculated sequentially rather than all at once, to preserve computer memory. The code runs in a loop, adding the information section by section to the CIF file.

4.1 FrankensteinCIF Full Function

Appendix B: DXF Code Repository

NOTE: All the code can be provided by request.

1. ExactGen

Modeled after CIFgen, ExactGen reads an input of photon sieve design parameters, and directly outputs a DXF file with the design. This program also displays information about the sieve, such as the radius, size of the smallest hole, and number of holes.

1.1 ExactGen Full Function

2. ExactVectors

ExactVectors takes a range of rings and writes the information for each circle to vectors. The function then repeats the process for the following range of rings until it reaches the end of the sieve.

2.1 ExactVectors Full Function

3. FrankensteinMonster

This function uses ExactVectors and HoleCounter to split the sieve into smaller sections. The hole information for each section is calculated sequentially rather than all at once, to preserve computer memory. The code runs in a loop, adding the information section by section to the DXF file.

3.1 FrankensteinMonster Full Function

4. ExactGen_Parts

This function, modeled after ExactGen, splits a DXF design into multiple files to accommodate file size limitations. The user is able to specify the number of files that the design is split into. In addition, this program displays the maximum hole size, minimum size, and overall diameter of each part.

4.1 ExactGen_Parts Full Function

5. PhotonSieve_Generation

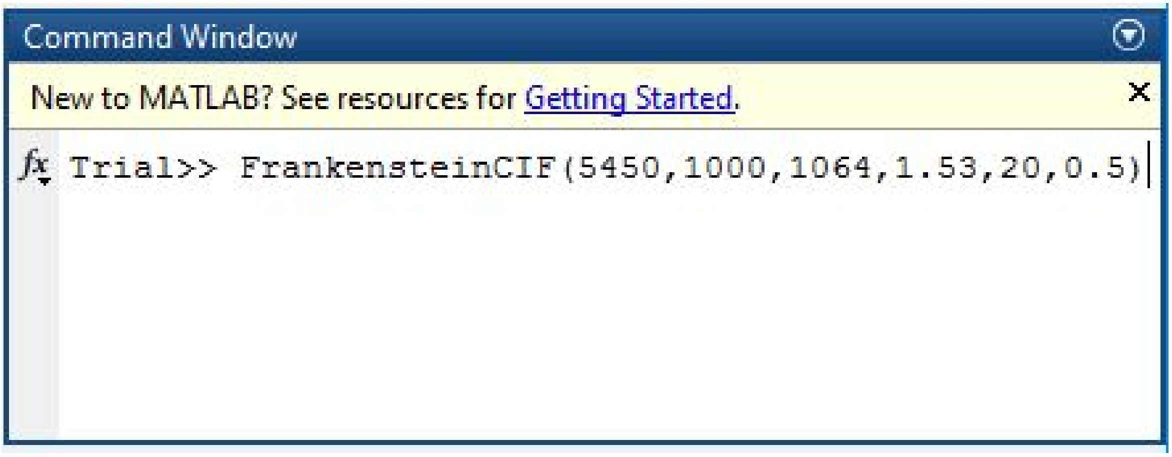
This function combines the other programs, and allows the user to create a photon sieve of either file type depending on the inputs. The other programs are nested within this one, so it can be transported easily.

5.1 PhotonSieve_Generation Full Function

Appendix C: PS Design and Verification Process

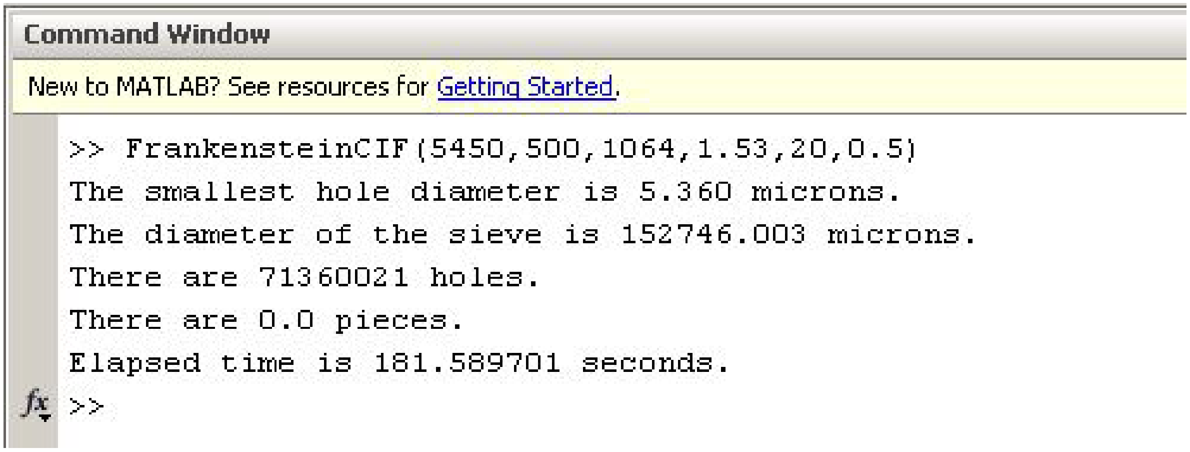
This appendix provides a step-by-step process to create the PS design and verify the size and information of this design using preview software.

1. Open MATLAB.
2. Choose the appropriate MATLAB function (based on file type desired), and enter function in MATLAB Command Window with desired PS parameters. Run the program.



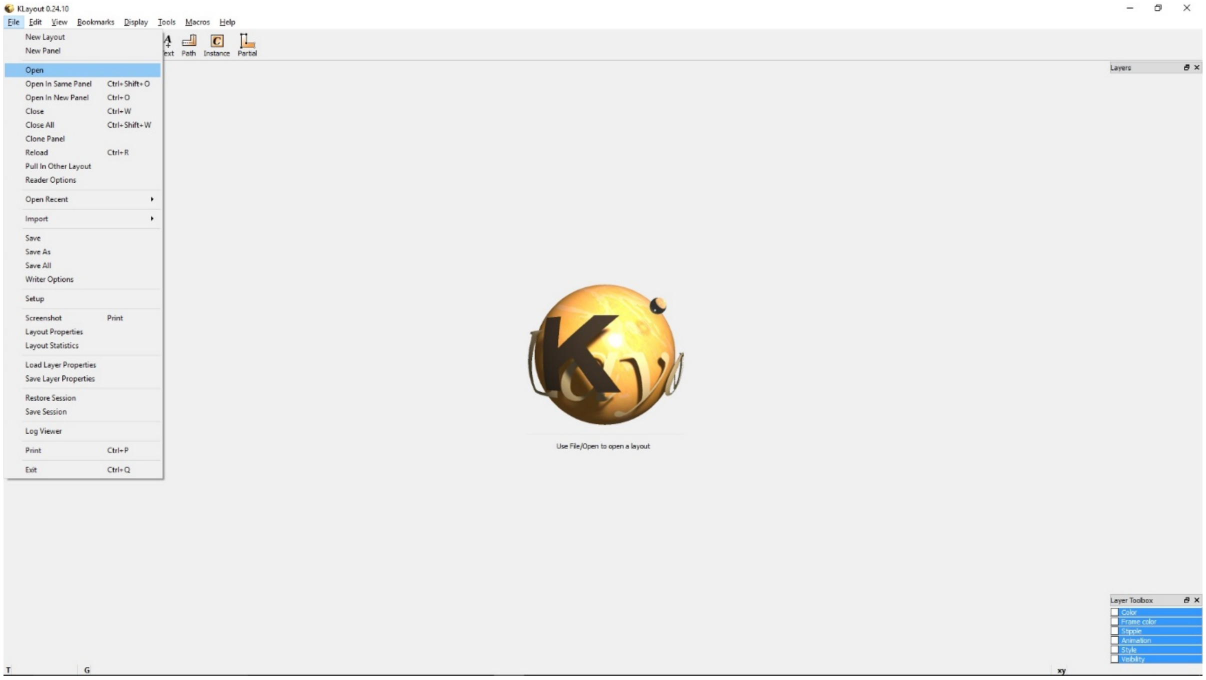
```
Trial>> FrankensteinCIF(5450,1000,1064,1.53,20,0.5)
```

3. View the Command Window output to determine that the sieve is the correct size and meets specifications.



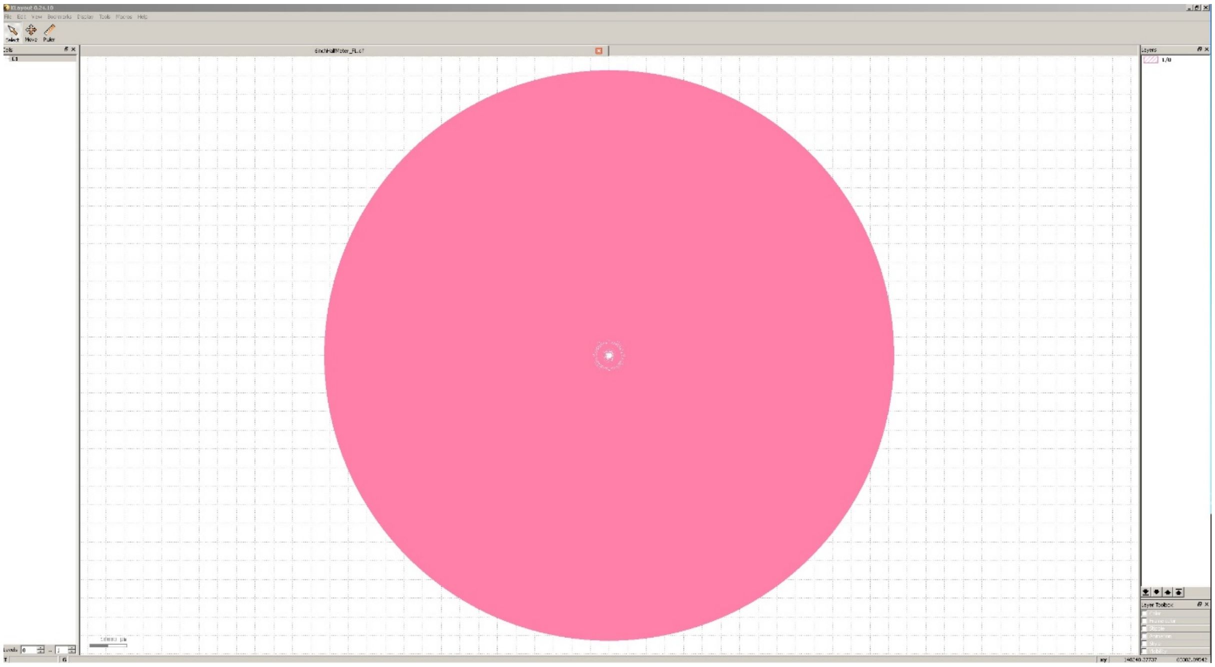
```
>> FrankensteinCIF(5450,500,1064,1.53,20,0.5)
The smallest hole diameter is 5.360 microns.
The diameter of the sieve is 152746.003 microns.
There are 71360021 holes.
There are 0.0 pieces.
Elapsed time is 181.589701 seconds.
>>
```

4. Open one of the preview programs described earlier. KLayout is recommended, and is shown in the following steps. Open the design, by clicking “Open” under the “File” tab.

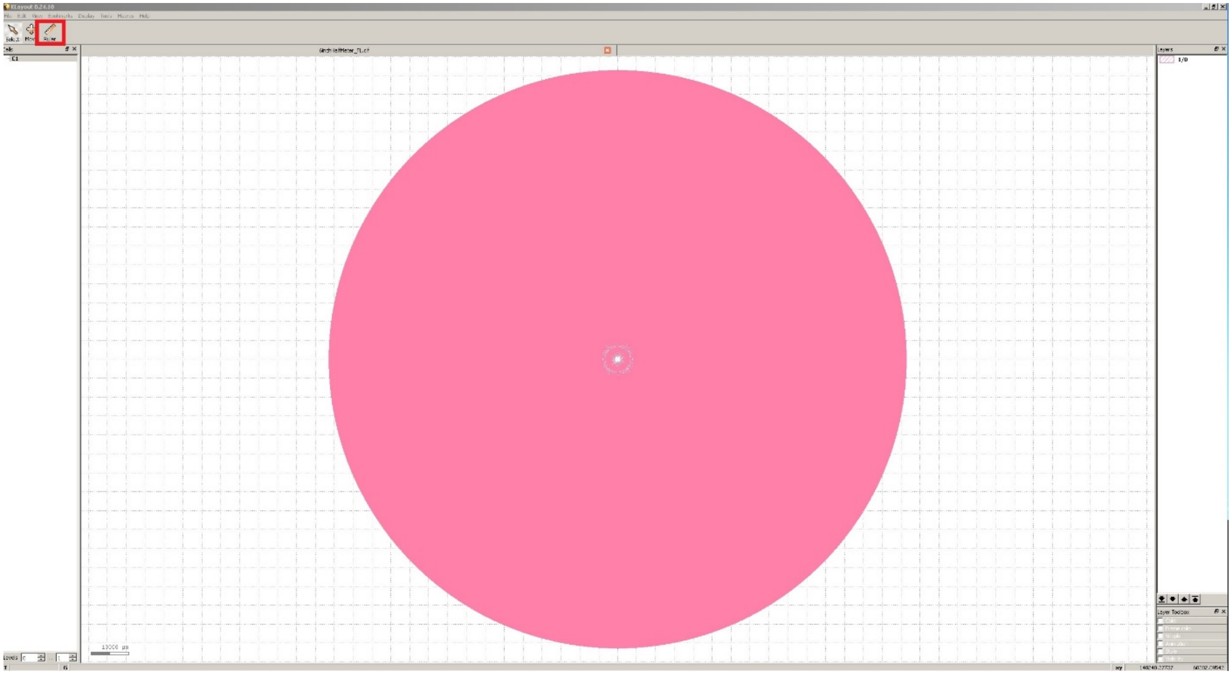


Use File/Open to open a layout

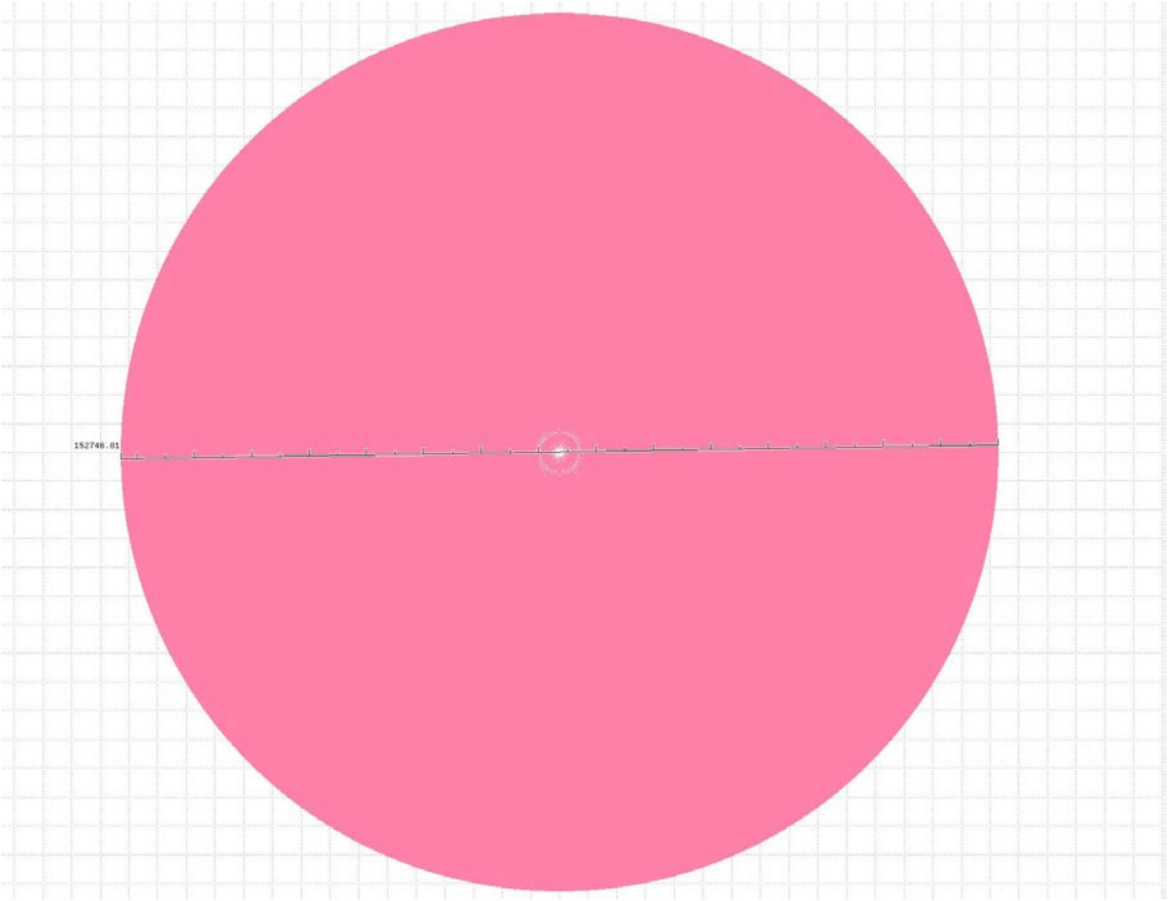
5. Wait until the design has fully loaded. Below is an example of a fully loaded design in the software.



6. To measure the overall sieve diameter, click the "Ruler" tool in the top left corner of the window.

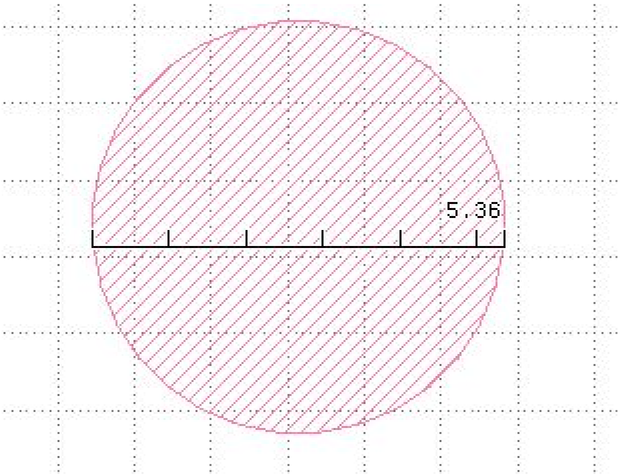


7. Click on a point on the edge of the sieve. Move the mouse to a point on the opposite side of the sieve. The line connecting the first point to the second should pass through the center of the sieve. Once this has been done, click the second point. The line should remain, with a measurement of the distance between the two points. This is the diameter of the sieve, and should match the Command Prompt value.

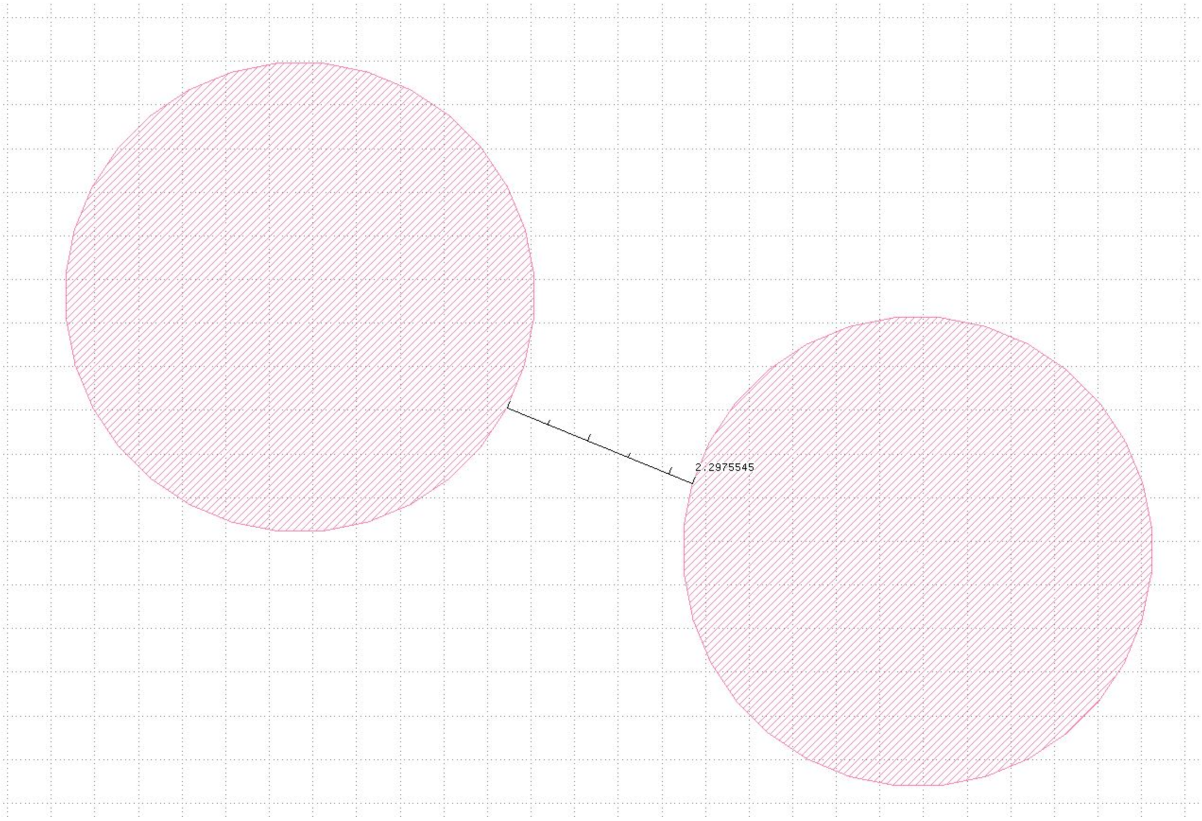


8. Pan to the edge of the sieve. This can be done by clicking the mouse wheel and moving the mouse. Then, zoom in to circles on the very edge of the sieve. Zoom by scrolling the mouse wheel.

9. Repeat step 7 to determine the diameter of the outermost (smallest) circle. This should match the value from the Command Window.



10. Using the same measurement technique, determine the distance between two adjacent holes. This ensures that the holes are far apart enough, depending on the head size used by the Heidelberg DWL 66fs.

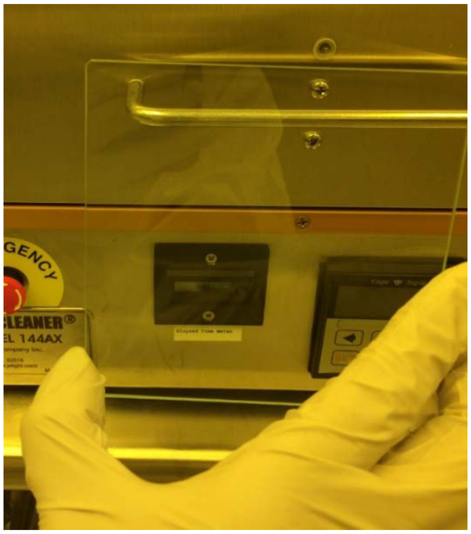


11. Exit KLayout. If the sieve meets desired specifications, it is ready to be manufactured. If the parameters need to be changed, repeat from step 2.

Appendix D: Step-by-Step process for 6 inch PS fabrication

1. Sample Preparation

1.1 Prepare transparent soda lime glass substrate (7” x 7” x 0.09” dimension)



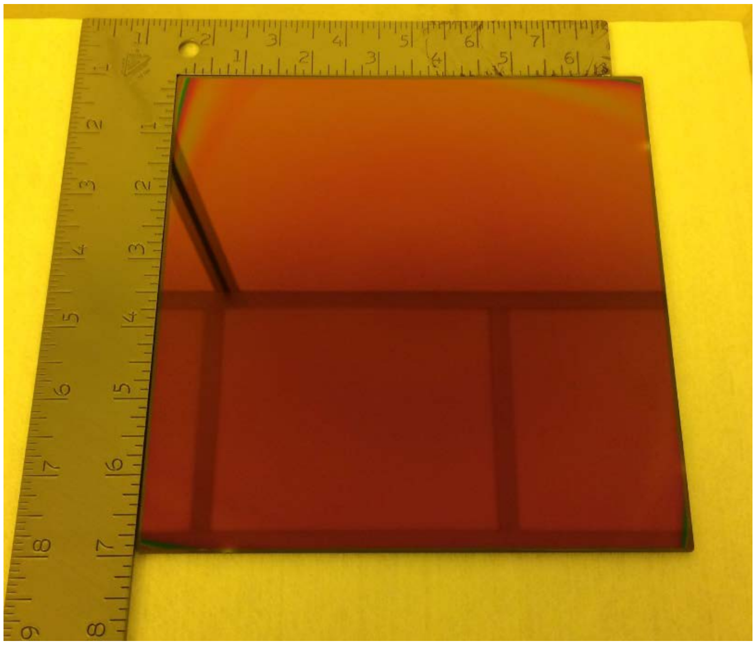
1.2 Clean the soda lime glass substrate (5 minutes acetone, 5 minutes isopropyl alcohol, and 10 minutes Deionized (DI) water).

1.3 Deposit 200 nm thick Chromium (Cr) on the cleaned glass substrate by using a magnetron sputtering system.

1.4 Deposit a 530 nm thick Photoresist (PR) coating on the Cr/soda lime glass substrate.

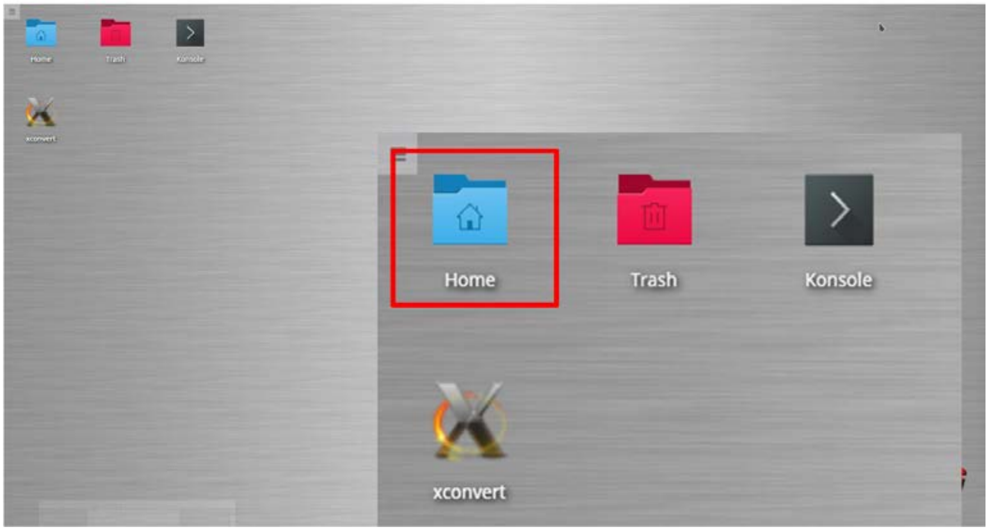
1.5 Pre-bake the PR coated Cr/soda lime glass substrate at 103°C for 30 minutes on the hot plate to remove the solvent.

1.6 The sample is ready for the laser patterning process using the Heidelberg laser (writer) lithography system.

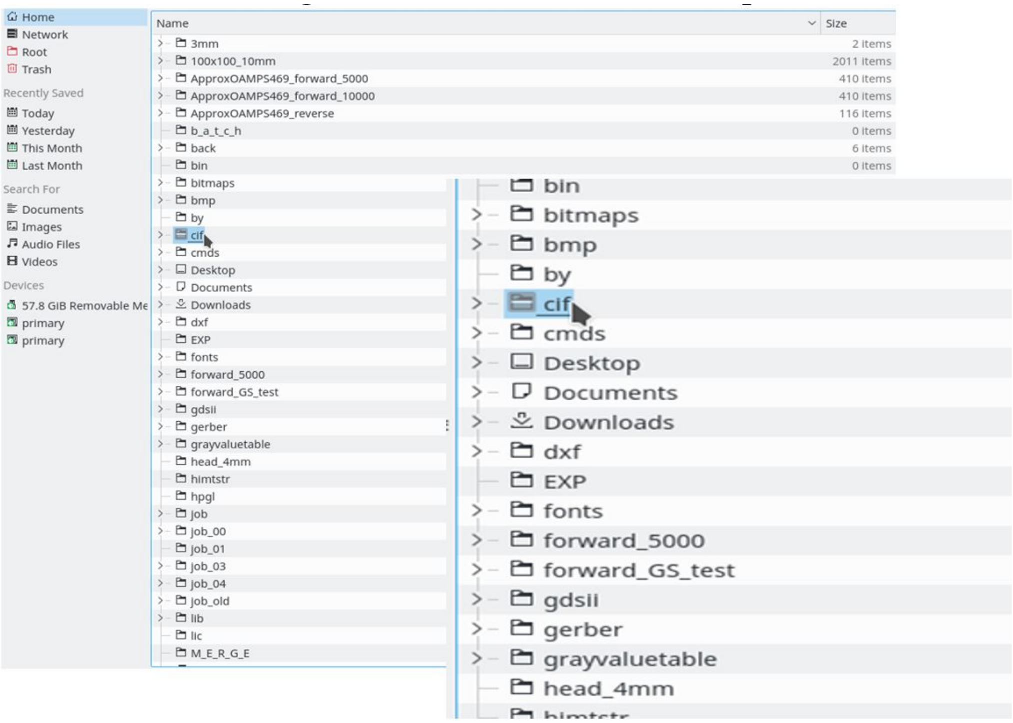


2. Data Conversion Process

2.1 Select the Home icon to transfer the mask file onto the Linux computer.

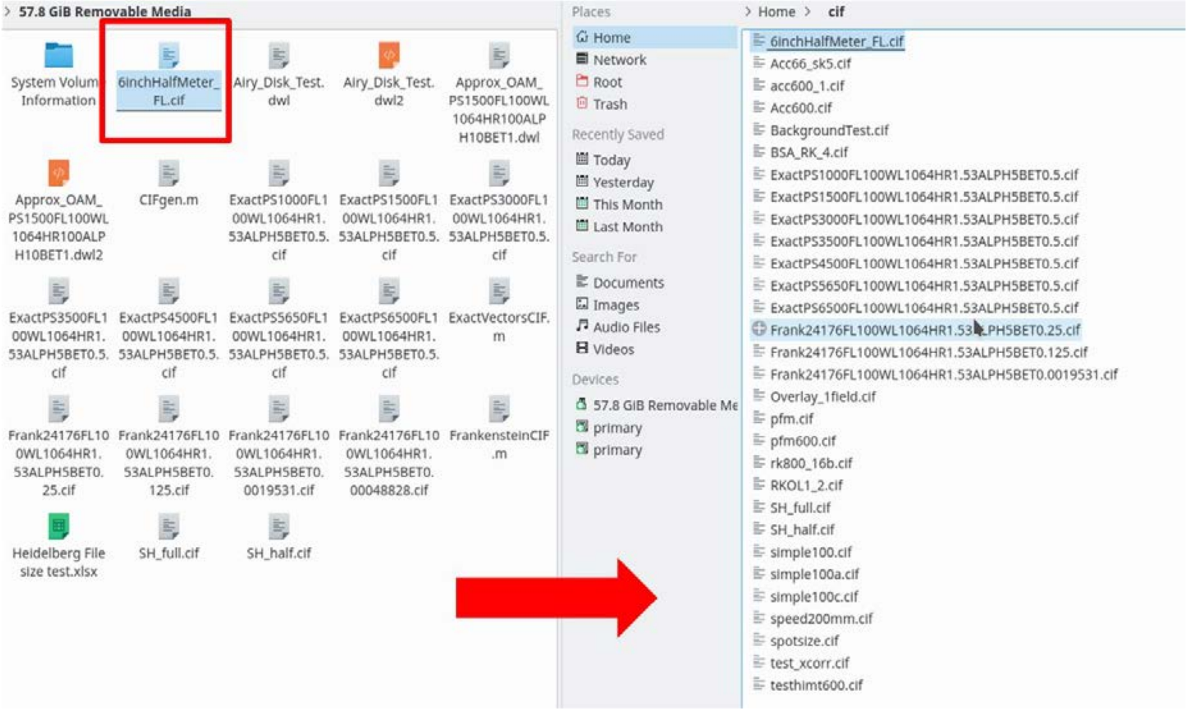


2.2 Copy the mask file over from the storage device onto the Linux computer



Home
Network
Root
Trash
Recently Saved
Today
Yesterday
This Month
Last Month
Search For
Documents
Images
Audio Files
Videos
Devices
57.8 GiB Removable Me
primary
primary

Name	Size
3mm	2 items
100x100_10mm	2011 items
ApproxOAMPS469_forward_5000	410 items
ApproxOAMPS469_forward_10000	410 items
ApproxOAMPS469_reverse	116 items
b_a_t_c_h	0 items
back	6 items
bin	0 items
bitmaps	
bmp	
by	
cif	
cmds	
Desktop	
Documents	
Downloads	
dxf	
EXP	
fonts	
forward_5000	
forward_GS_test	
gdsii	
gerber	
grayvaluetable	
head_4mm	
himtstr	
hpgl	
job	
job_00	
job_01	
job_03	
job_04	
job_old	
lib	
lic	
M_E_R_G_E	



57.8 GiB Removable Media

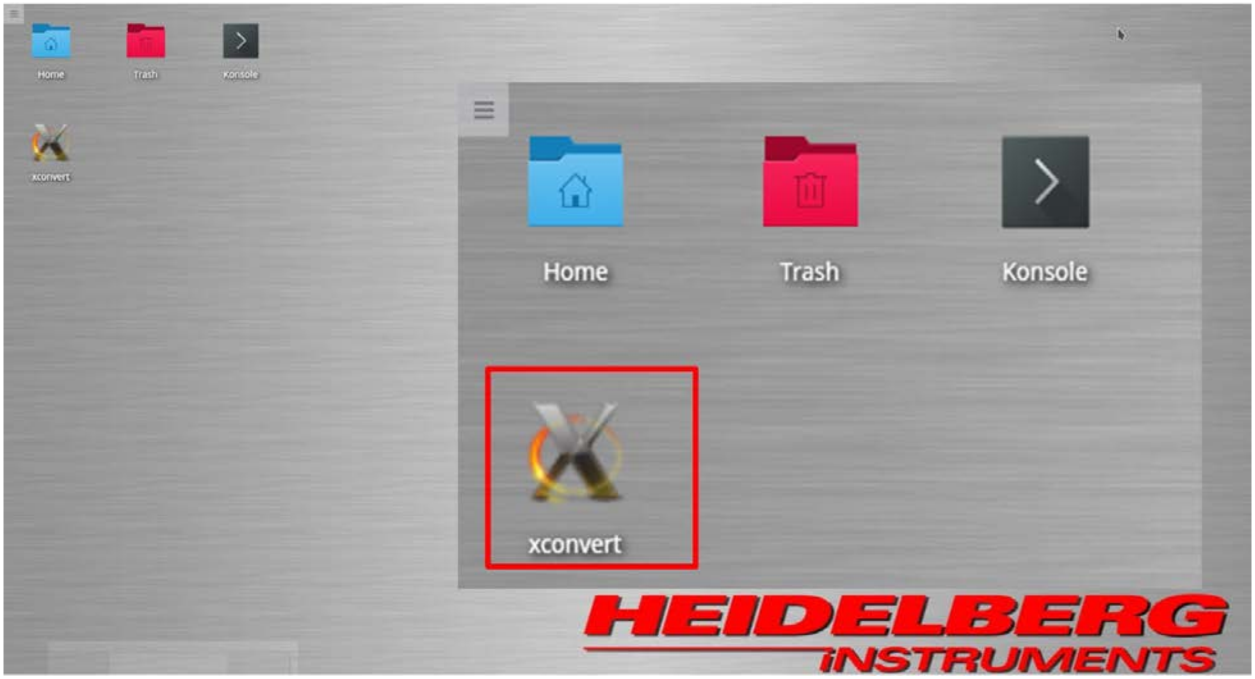
System Volume Information
6inchHalfMeter_FL.cif
Airy_Disk_Test.dwl
Airy_Disk_Test.dwl2
Approx_OAMPS1500FL100WL1064HR100ALPH10BET1.dwl
Approx_OAMPS1500FL100WL1064HR100ALPH10BET1.dwl2
CIFgen.m
ExactPS1000FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS1500FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS3000FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS3500FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS4500FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS5650FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS6500FL100WL1064HR1.53ALPH5BET0.5.cif
ExactVectorsCIF.m
Frank24176FL100WL1064HR1.53ALPH5BET0.25.cif
Frank24176FL100WL1064HR1.53ALPH5BET0.125.cif
Frank24176FL100WL1064HR1.53ALPH5BET0.0019531.cif
Frank24176FL100WL1064HR1.53ALPH5BET0.00048828.cif
FrankensteinCIF.m
Heidelberg File size test.xlsx
SH_full.cif
SH_half.cif

Places
Home
Network
Root
Trash
Recently Saved
Today
Yesterday
This Month
Last Month
Search For
Documents
Images
Audio Files
Videos
Devices
57.8 GiB Removable Me
primary
primary

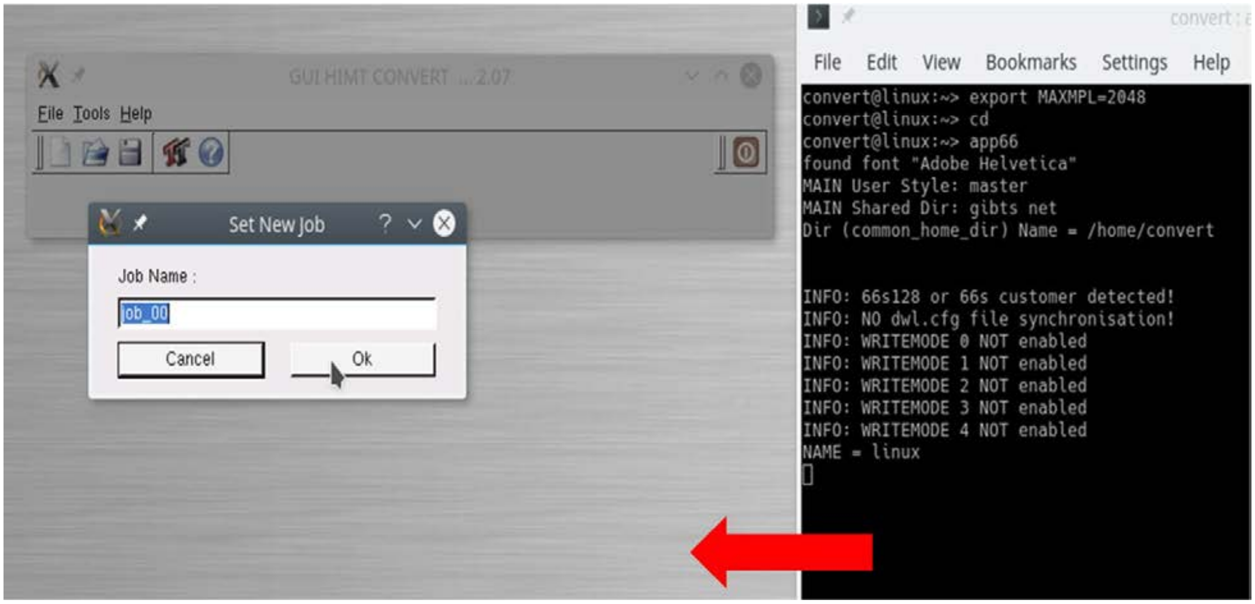
Home > cif

6inchHalfMeter_FL.cif
Acc66_sk5.cif
acc600_1.cif
Acc600.cif
BackgroundTest.cif
BSA_RK_4.cif
ExactPS1000FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS1500FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS3000FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS3500FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS4500FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS5650FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS6500FL100WL1064HR1.53ALPH5BET0.5.cif
Frank24176FL100WL1064HR1.53ALPH5BET0.25.cif
Frank24176FL100WL1064HR1.53ALPH5BET0.125.cif
Frank24176FL100WL1064HR1.53ALPH5BET0.0019531.cif
Overlay_1field.cif
pfm.cif
pfm600.cif
rk800_16b.cif
RKOL1_2.cif
SH_full.cif
SH_half.cif
simple100.cif
simple100a.cif
simple100c.cif
speed200mm.cif
spotsize.cif
test_xcorr.cif
testhimt600.cif

2.3 Double click the “xconvert” icon to begin the conversion process.



2.4 Open a new job.


GUI HIMT CONVERT ... 2.07
File Tools Help
Set New Job
Cancel
Ok
File Edit View Bookmarks Settings Help

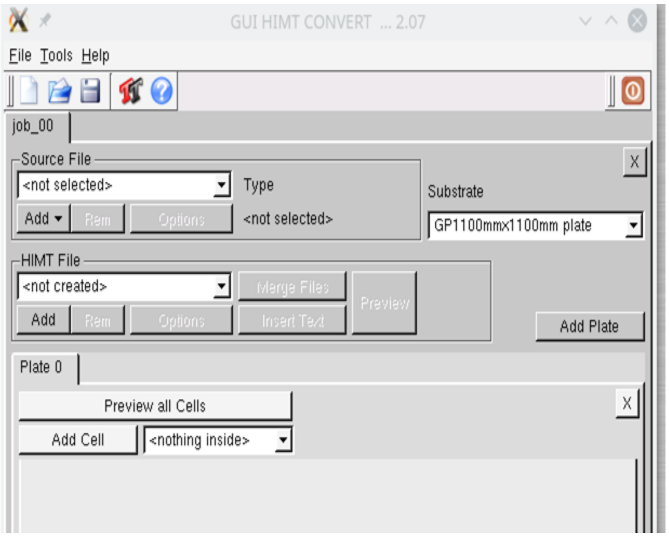
Job Name : job_00

```
convert@linux:~> export MAXMPL=2048
convert@linux:~> cd
convert@linux:~> app66
found font "Adobe Helvetica"
MAIN User Style: master
MAIN Shared Dir: gibts net
Dir (common_home_dir) Name = /home/convert


INFO: 66s128 or 66s customer detected!
INFO: NO dwl.cfg file synchronisation!
INFO: WRITEMODE 0 NOT enabled
INFO: WRITEMODE 1 NOT enabled
INFO: WRITEMODE 2 NOT enabled
INFO: WRITEMODE 3 NOT enabled
INFO: WRITEMODE 4 NOT enabled
NAME = linux
```

2.5 Title the new job whatever you deem appropriate.

2.6 The conversion window will appear.



Source File <not selected>

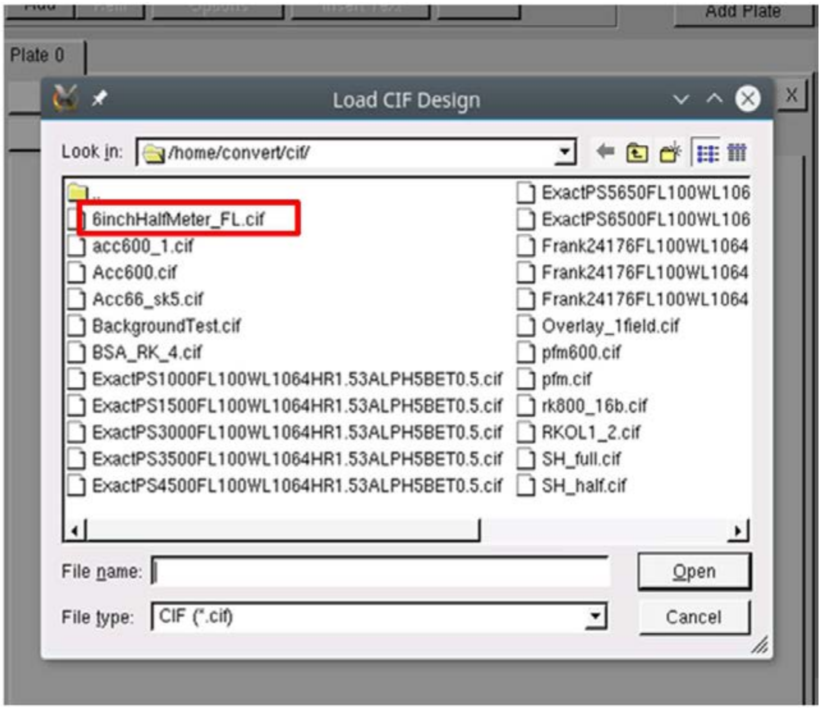
Type <not selected>

Substrate GP1100mmx1100mm plate

HIMT File <not created>

Add Cell <nothing inside>

2.7 Use the Add drop-down box under “Source File” to select your file type (.cif) and subsequently your file (6inchHalfMeter.cif).

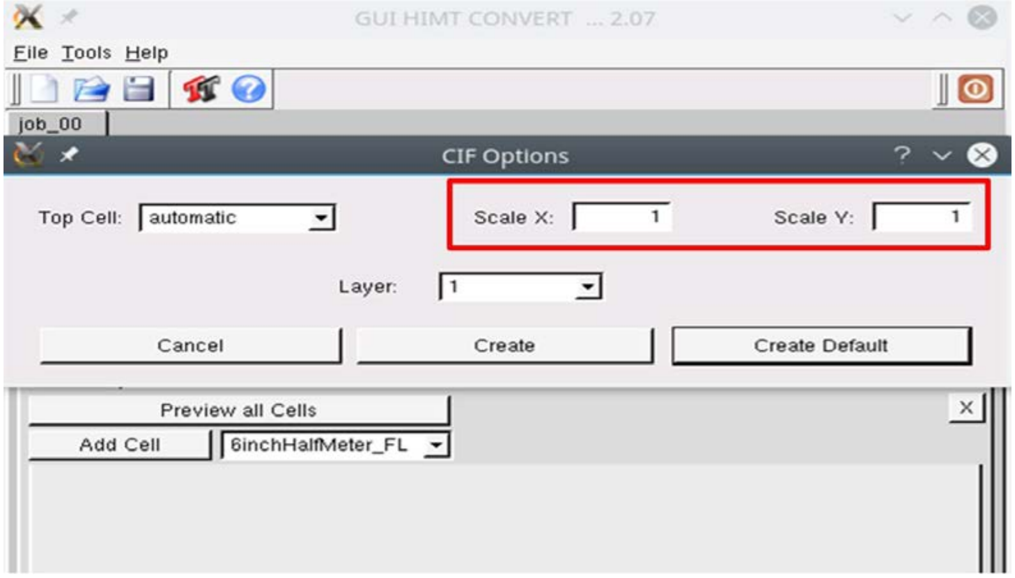


Look in: /home/convert/cif/

..
6inchHalfMeter_FL.cif
acc600_1.cif
Acc600.cif
Acc66_sk5.cif
BackgroundTest.cif
BSA_RK_4.cif
ExactPS1000FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS1500FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS3000FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS3500FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS4500FL100WL1064HR1.53ALPH5BET0.5.cif
ExactPS5650FL100WL106
ExactPS6500FL100WL106
Frank24176FL100WL1064
Frank24176FL100WL1064
Frank24176FL100WL1064
Overlay_1field.cif
pfm600.cif
pfm.cif
rk800_16b.cif
RKOL1_2.cif
SH_full.cif
SH_half.cif

File type: CIF (*.cif)

2.8 This will bring up your options screen. For a .cif file, make sure that the magnification factor can be used to scale all of the features by a common factor. When you are finished selecting the options, select the “Create Default” button, which will save your selections under the same job name you have been using.



Top Cell: automatic

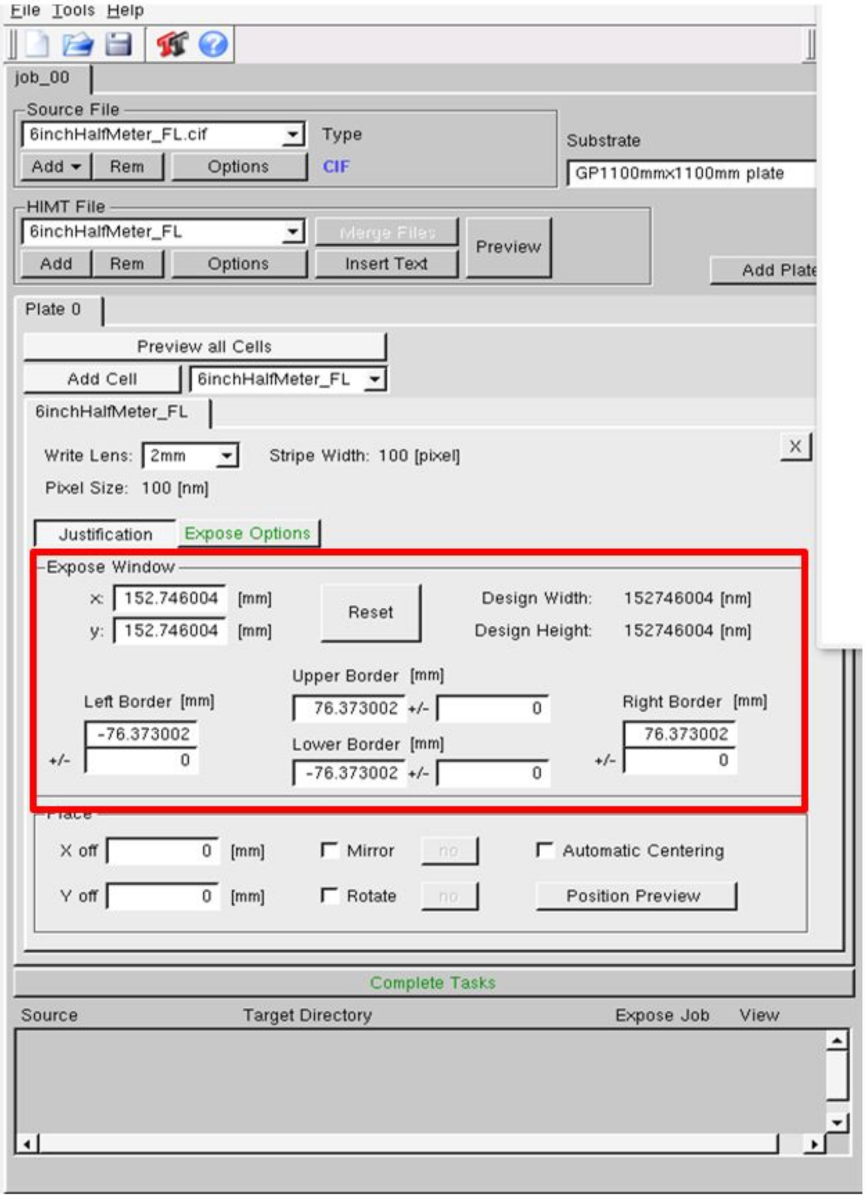
Scale X: 1

Scale Y: 1

Layer: 1

6inchHalfMeter_FL

2.9 The expose window options allows you to choose exactly how much of your selected file should be exposed. The default settings will expose your entire mask file, so if that is your intent, these setting should be left alone. However, if a smaller window of your entire mask is all that you want to expose, select the appropriate borders and check your settings in the preview window.



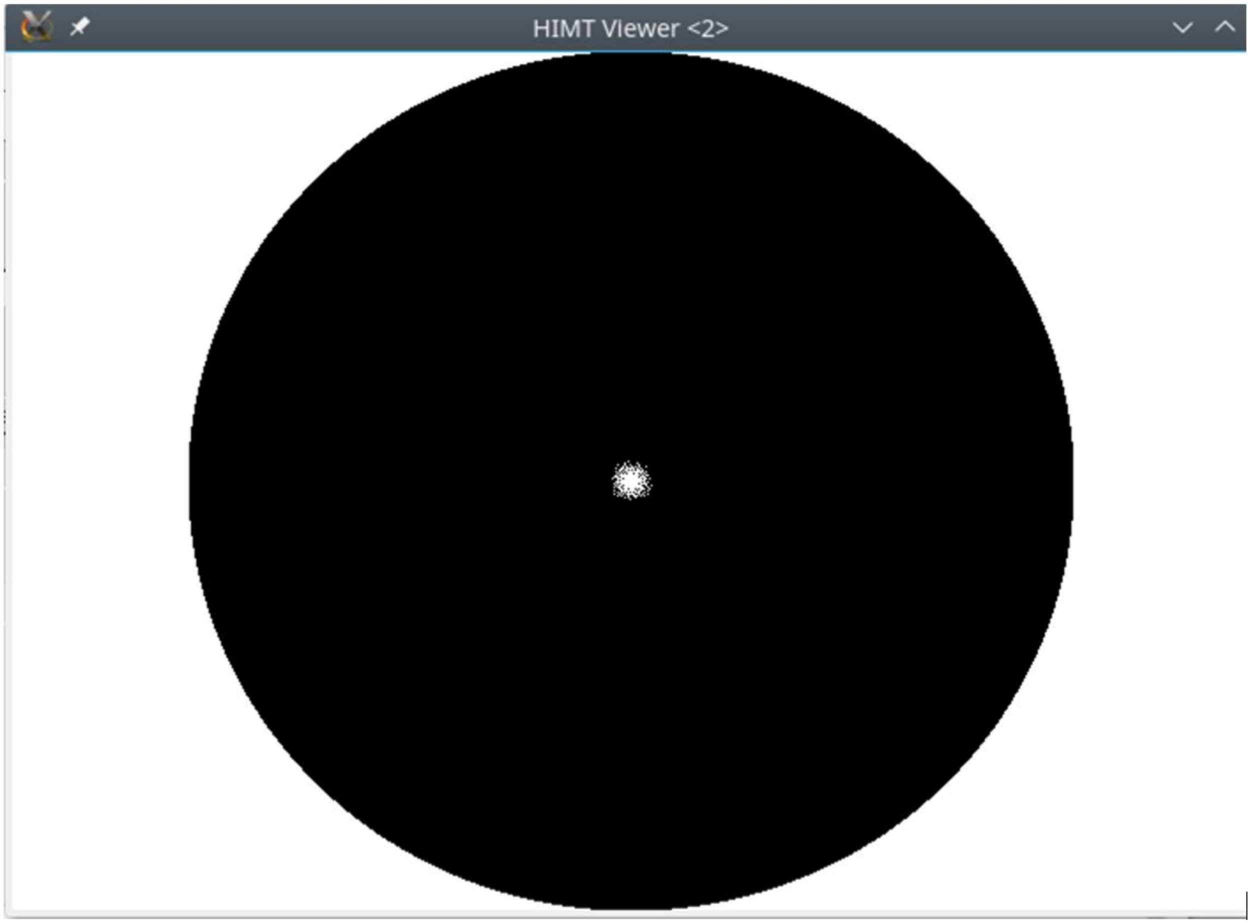
Source File: 6inchHalfMeter_FL.cif
Type: CIF
Substrate: GP1100mmx1100mm plate
HIMT File: 6inchHalfMeter_FL
Add Cell: 6inchHalfMeter_FL

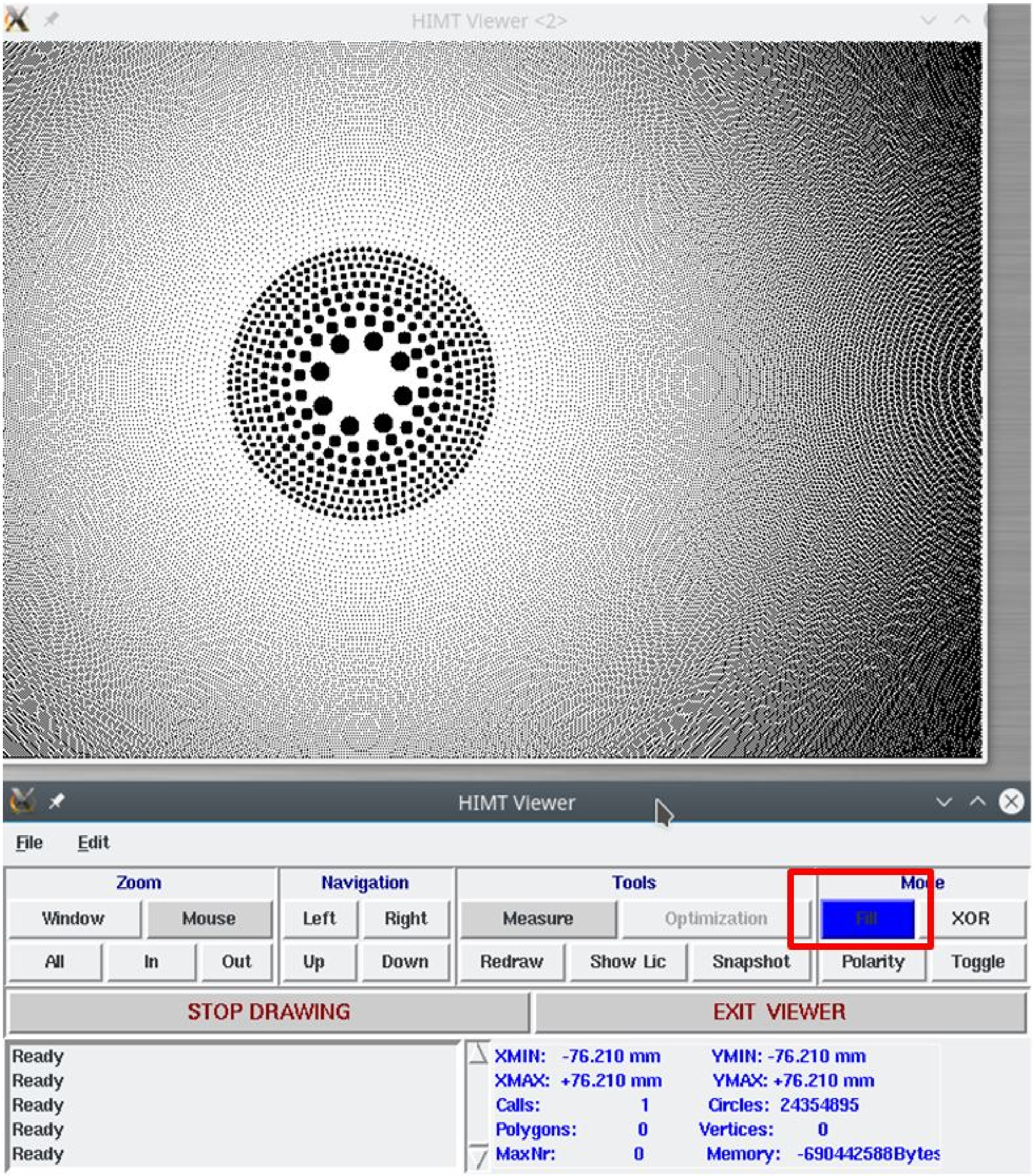
6inchHalfMeter_FL
Write Lens: 2mm Stripe Width: 100 [pixel]
Pixel Size: 100 [nm]

Expose Window
x: 152.746004 [mm] Design Width: 152746004 [nm]
y: 152.746004 [mm] Design Height: 152746004 [nm]
Upper Border [mm]: 76.373002 +/- 0
Left Border [mm]: -76.373002 +/- 0
Lower Border [mm]: -76.373002 +/- 0
Right Border [mm]: 76.373002 +/- 0

Place
X off 0 [mm] □ Mirror no □ Automatic Centering
Y off 0 [mm] □ Rotate no

2.10 Click the Preview button located in the HIMT filed to preview your selections to assure it is what you intended. This will bring up the HIMT Viewer which will allow you to zoom in and examine what you have selected. If this is not correct, it is advised to return to step 8 and change the structure or the layer if either of those is incorrect. Select Exit Viewer when you are finished.



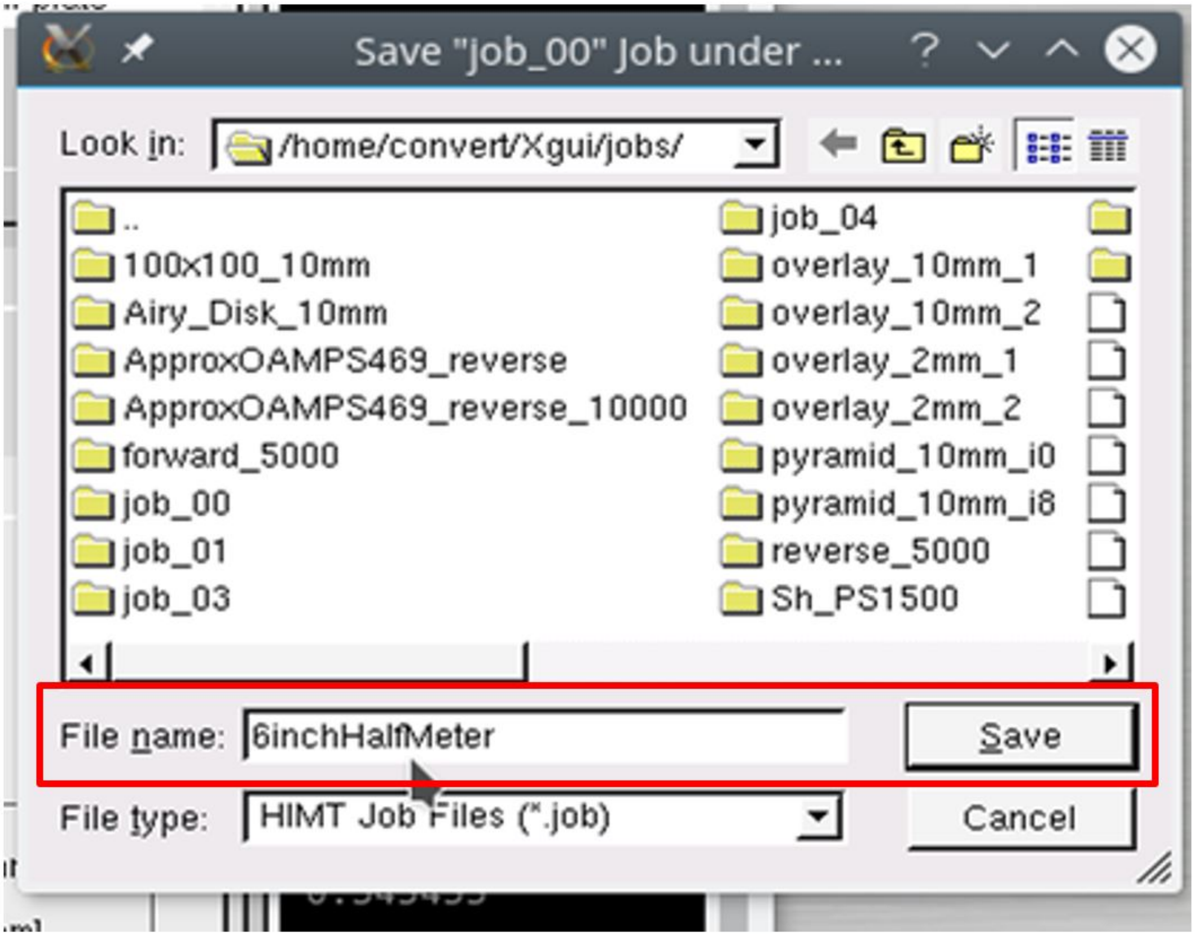


```
Ready
Ready
Ready
Ready
Ready
```

```
XMIN:  -76.210 mm        YMIN: -76.210 mm
XMAX:  +76.210 mm        YMAX: +76.210 mm
Calls:            1      Circles:  24354895
Polygons:         0      Vertices:       0
MaxNr:            0      Memory:  -690442588Bytes
```

2.11 The “Place” setting allows the user to offset the design from center. Check the automatic centering option which will allow the computer to find the geometric center of your mask. It will then place that point at the exact center of the mask it is writing.

2.12 Save under the desired name (“6inchHalfMeter”).



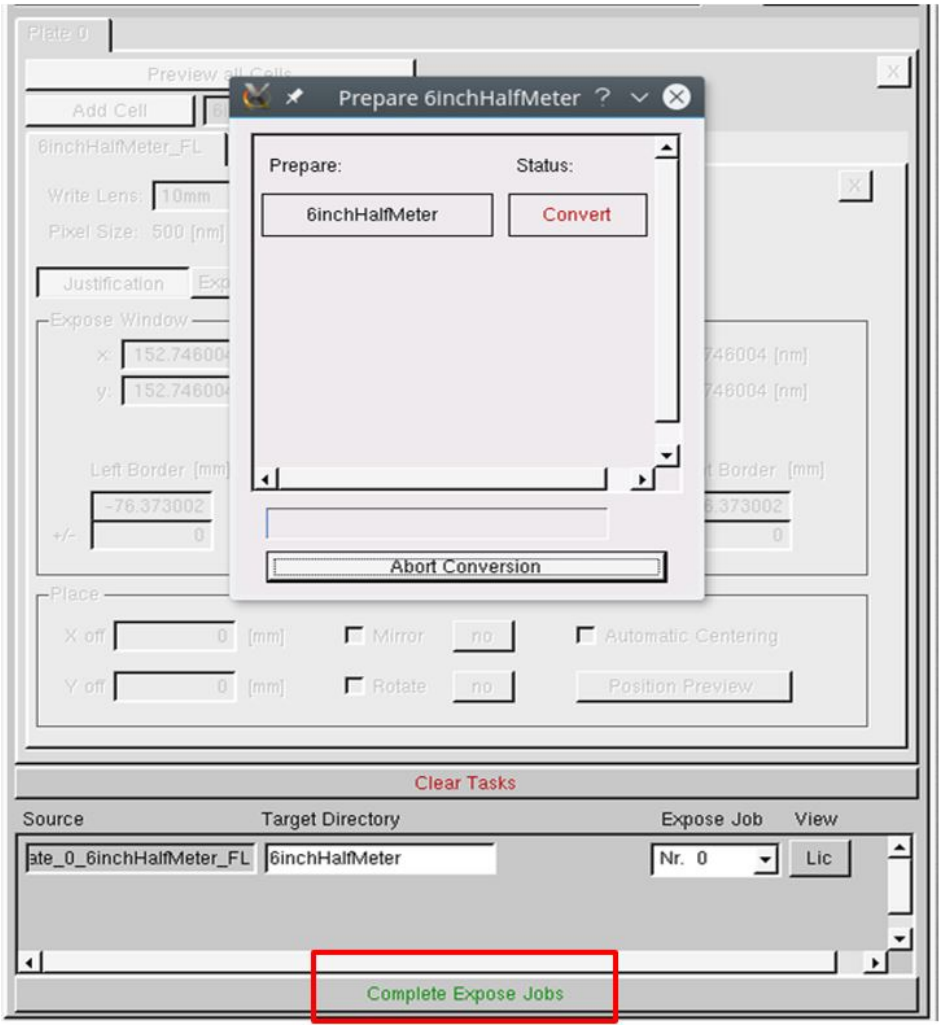
Save "job_00" Job under ...

Look in: /home/convert/Xgui/jobs/

File name: 6inchHalfMeter

File type: HIMT Job Files (*.job)

2.13 Click the “Complete Expose Jobs” button.



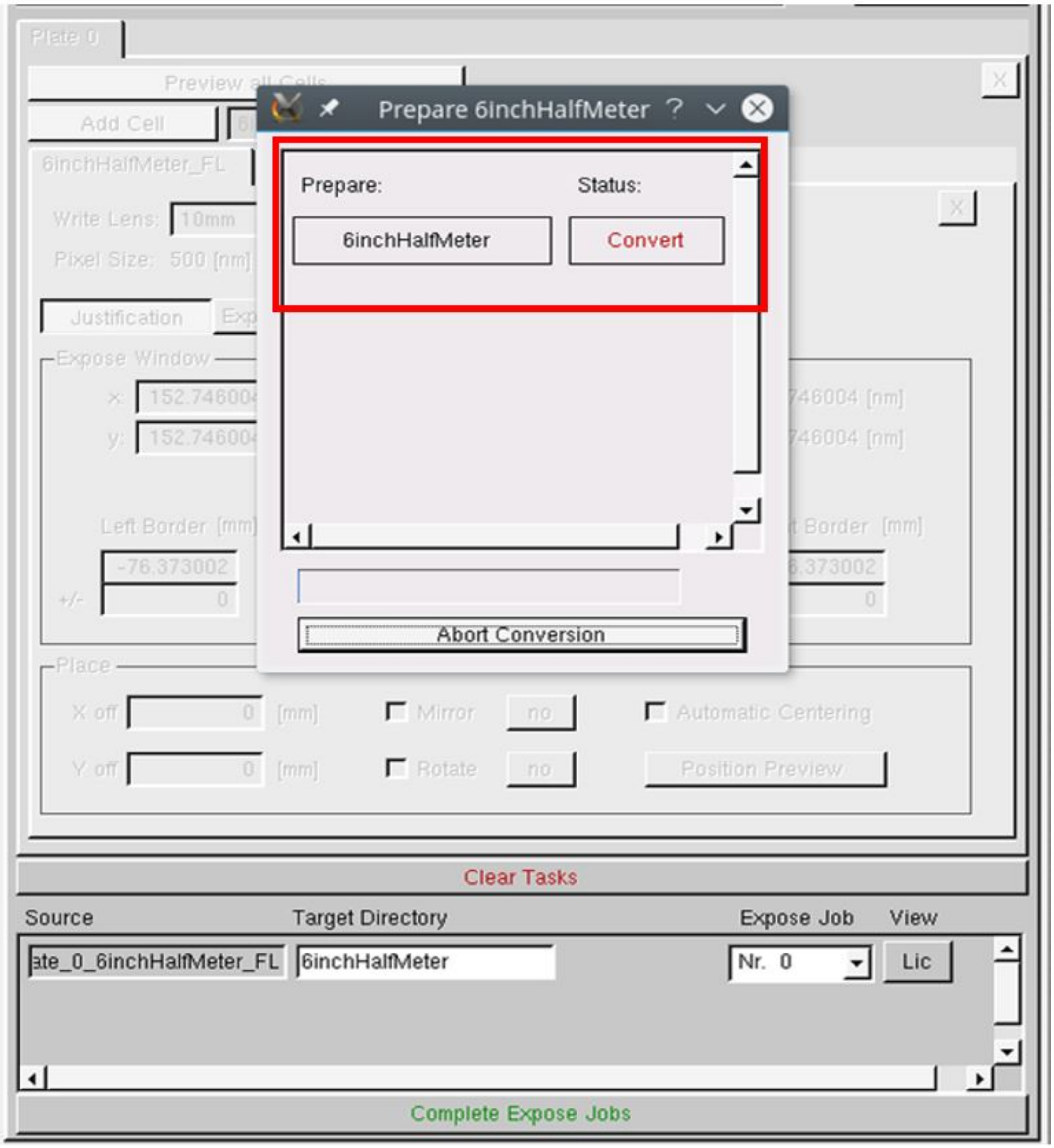
Prepare 6inchHalfMeter

Prepare: 6inchHalfMeter

Status: Convert

Source	Target Directory	Expose Job	View
ate_0_6inchHalfMeter_FL	6inchHalfMeter	Nr. 0	Lic

2.14 Automatically start the conversion process. Do not click the Abort Conversion button unless you need to stop the conversion process.


Prepare 6inchHalfMeter

Prepare: 6inchHalfMeter

Status: Convert

Write Lens: 10mm

Pixel Size: 500 [nm]

x: 152.74600

y: 152.74600

Left Border [mm] -76.373002

+/- 0

X off 0 [mm]

Y off 0 [mm]

Mirror no

Rotate no

Source: ate_0_6inchHalfMeter_FL

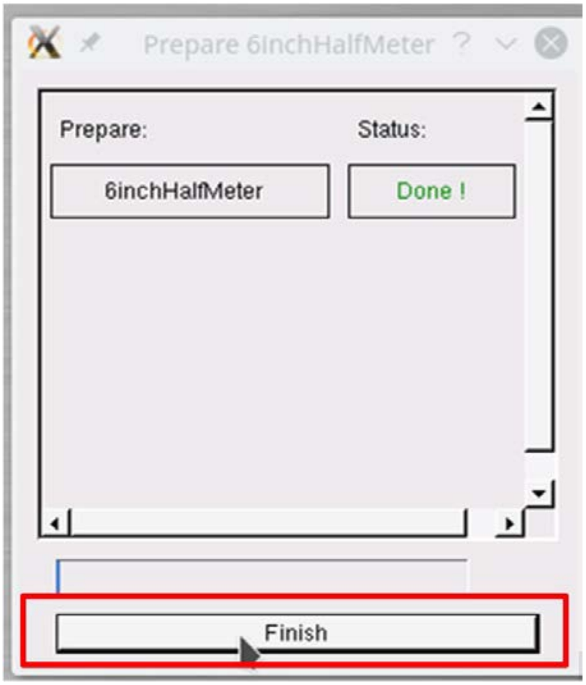
Target Directory: 6inchHalfMeter

Expose Job: Nr. 0

View: Lic

2.15 During the conversion process, 3,055 lic files are generated from the “6inchHalfMeter” file. Each lic file results from one scan. When either the 2mm or the 10mm head are selected, the pixel size will be displayed, which can be used for calculations. For the 10mm head, the size is 100 pixels for one scan, and the pixel width is 500 nm. Therefore, each scan will be 100 pixels x 500 nm (the pixel width), which is 50 µm. The 6” x 6” design dimension has dimensions of 152.746 mm x 152.746 mm. Thus, 152,746 µm (the total width) divided by 50 µm (the scan width), will result in 3,055 lic files.

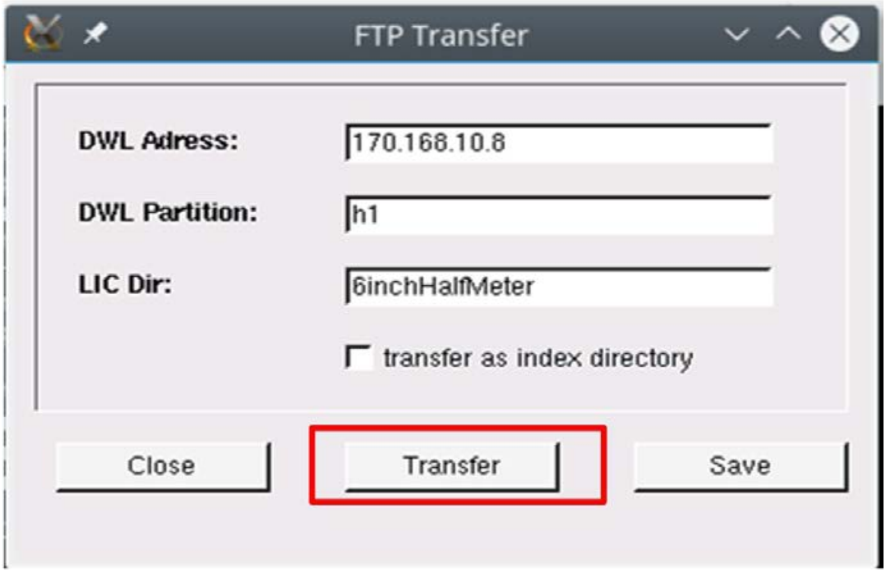
2.16 Hit finish once the conversion is complete and the FTP transfer process will start.



Prepare: 6inchHalfMeter

Status: Done !

2.17 Complete the FTP transfer under default settings, and once the file is completed, simply Save and Exit. No notification is given when the file finishes transferring, the only change is that the transfer button is no longer depressed.



DWL Adress: 170.168.10.8

DWL Partition: h1

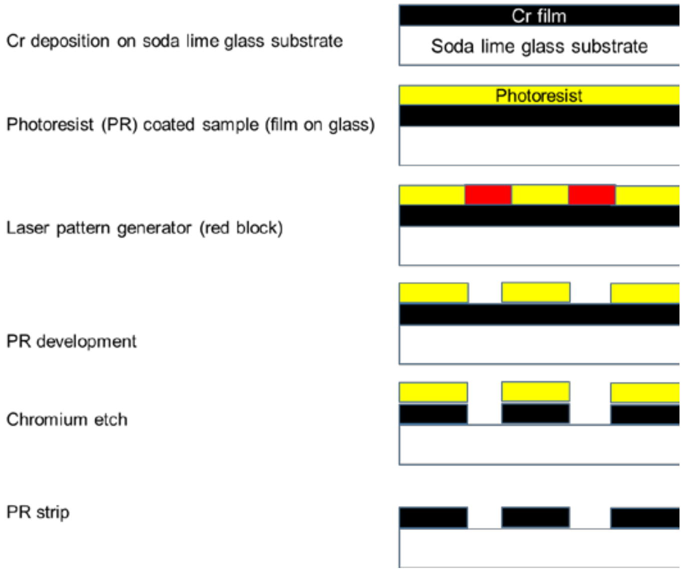
LIC Dir: 6inchHalfMeter

□ transfer as index directory

2.18 Clear Tasks and exit the program.

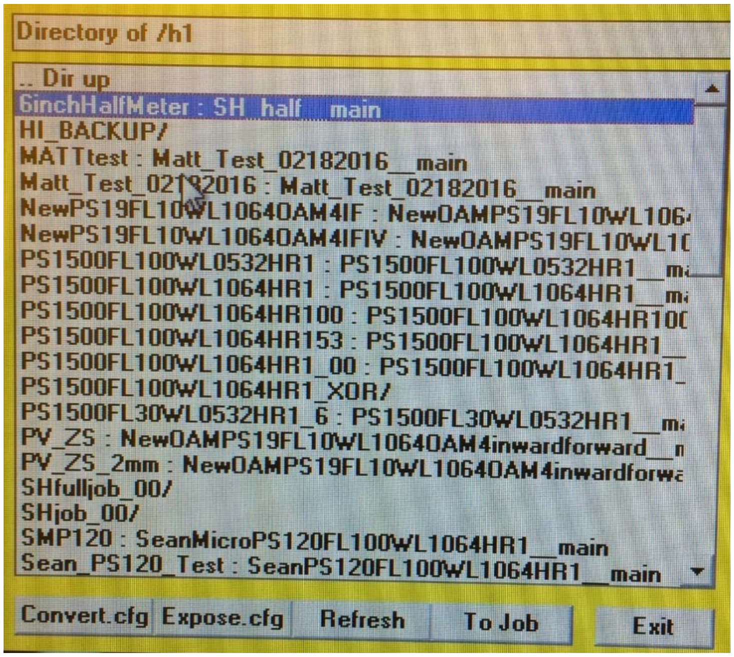
3. Heidelberg Laser Lithography Process

Following is a flow chart summarizing the lithography process and the post-lithography process.


Cr deposition on soda lime glass substrate
Cr film
Soda lime glass substrate
Photoresist (PR) coated sample (film on glass)
Photoresist
Laser pattern generator (red block)
PR development
Chromium etch
PR strip

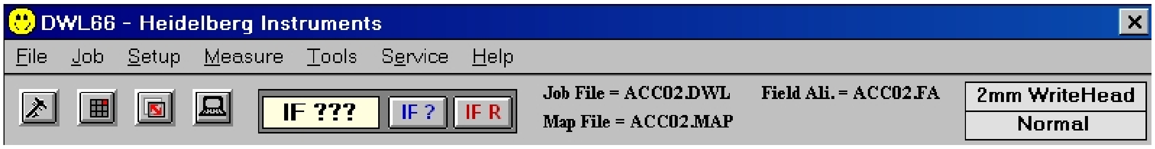
3.1 On the Windows PC, open the DWL software.

3.2 Load the configuration file (file name: “6inchHalfMeter”) for the write-head that will be used for exposure.



```
Directory of /h1
.. Dir up
6inchHalfMeter : SH_half__main
HI_BACKUP/
MATTtest : Matt_Test_02182016__main
Matt_Test_02182016 : Matt_Test_02182016__main
NewPS19FL10WL10640AM4IF : NewOAMPS19FL10WL106
NewPS19FL10WL10640AM4IFIV : NewOAMPS19FL10WL1
PS1500FL100WL0532HR1 : PS1500FL100WL0532HR1__m
PS1500FL100WL1064HR1 : PS1500FL100WL1064HR1__m
PS1500FL100WL1064HR100 : PS1500FL100WL1064HR10
PS1500FL100WL1064HR153 : PS1500FL100WL1064HR1_
PS1500FL100WL1064HR1_00 : PS1500FL100WL1064HR1_
PS1500FL100WL1064HR1_XOR/
PS1500FL30WL0532HR1_6 : PS1500FL30WL0532HR1__m
PV_ZS : NewOAMPS19FL10WL10640AM4inwardforward__n
PV_ZS_2mm : NewOAMPS19FL10WL10640AM4inwardforwa
SHfulljob_00/
SHjob_00/
SMP120 : SeanMicroPS120FL100WL1064HR1__main
Sean_PS120_Test : SeanPS120FL100WL1064HR1__main
```

3.3 Create the Map file. Go to **SETUP – NEW** in the main menu.



Job File = ACC02.DWL

Field Ali. = ACC02.FA

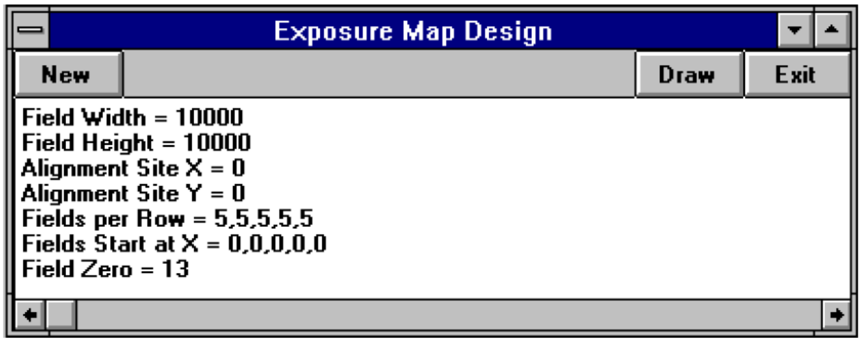
Map File = ACC02.MAP

3.4 The New Exposure Map window will appear. Click **Create Map** and input an eight character or less name that will be used as the project name for all exposure components.

3.5 Confirm the change to the new environment in the dialog box. You will notice 2 files are then created: a map file and a field alignment file.

3.6 Go to SETUP – EXPOSURE MAP. The exposure map design window will appear. Ensure that the field width and height are bigger than the design.

- Line 1 – Field width (microns). Must be large enough for design
- Line 2 – Field height (microns). Must be large enough for design
- Line 3/4 – Distance of alignment sites from origin (blank mask – leave default values)
- Line 5 – Fields per row – for each row of dies, enter the number of dies it should contain separated with commas.
- Line 6 – Fields start at X – position in microns where each row is to begin in the x-direction with respect to field furthest to the left.
- Line 7 – enter the field that contains the substrate origin.

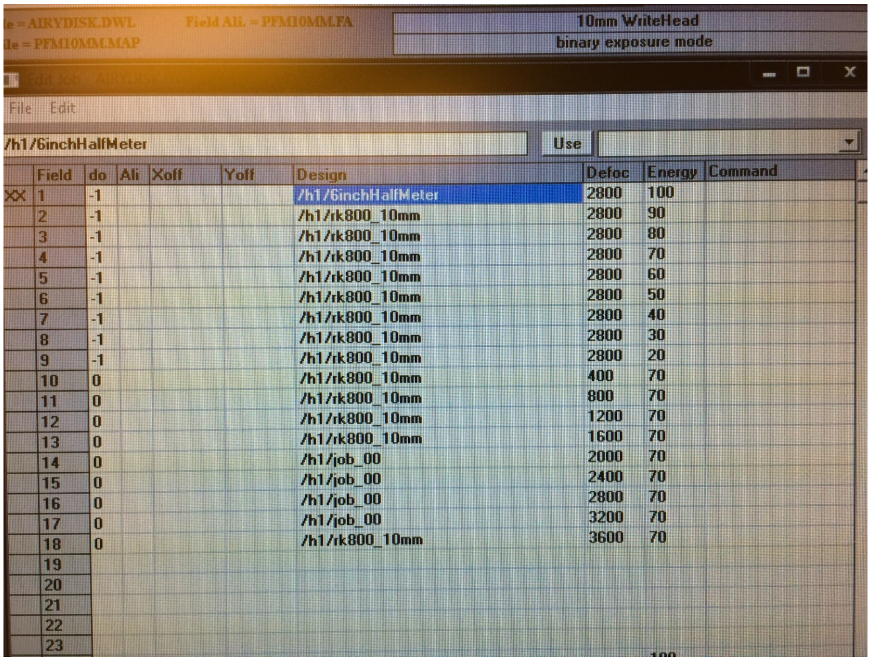


Field Width = 10000
Field Height = 10000
Alignment Site X = 0
Alignment Site Y = 0
Fields per Row = 5,5,5,5,5
Fields Start at X = 0,0,0,0,0
Field Zero = 13

3.7 Click **DRAW** to see the edited map.

3.8 Close the exposure map by pressing **EXIT**.

3.9 Now it is time to create the job file. Go to **JOB – MAKE JOB**. The "edit job" spreadsheet will open. This sheet allows users to choose exposure conditions.



le = AIRYDISK.DWL Field Ali. = PFM10MM.FA
le = PFM10MM.MAP

/h1/6inchHalfMeter

	Field	do	Ali	Xoff	Yoff	Design	Defoc	Energy	Command
XX	1	-1				/h1/6inchHalfMeter	2800	100	
	2	-1				/h1/rk800_10mm	2800	90	
	3	-1				/h1/rk800_10mm	2800	80	
	4	-1				/h1/rk800_10mm	2800	70	
	5	-1				/h1/rk800_10mm	2800	60	
	6	-1				/h1/rk800_10mm	2800	50	
	7	-1				/h1/rk800_10mm	2800	40	
	8	-1				/h1/rk800_10mm	2800	30	
	9	-1				/h1/rk800_10mm	2800	20	
	10	0				/h1/rk800_10mm	400	70	
	11	0				/h1/rk800_10mm	800	70	
	12	0				/h1/rk800_10mm	1200	70	
	13	0				/h1/rk800_10mm	1600	70	
	14	0				/h1/job_00	2000	70	
	15	0				/h1/job_00	2400	70	
	16	0				/h1/job_00	2800	70	
	17	0				/h1/job_00	3200	70	
	18	0				/h1/rk800_10mm	3600	70	
	19								
	20								
	21								
	22								
	23								

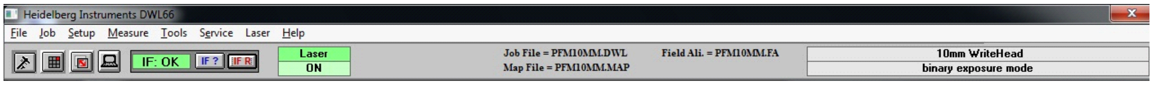
- “do” column – leave blank if the field is NOT to be written. Place a “-1” to write that field.
- “Design” – path to your design(s) converted. Click on the first row of design column. In the main menu, go to **FILE – DESIGNS.** The design directory will appear. Find the design you converted (File name: 6inchHalfMeter) and double click (or click “To do”) so that the path appears in the design field spreadsheet.
- “Defoc” – enter appropriate value (see write-head parameters below)
- “Energy” – enter appropriate value (see write-head parameters below)

- 10mm write head parameters for photomask (updated parameter on 08/01/2017)

Write-head	Approximate resolution limit	Filters	Defocus	Energy
10mm	1.2 microns	30% + 50%	2800	70

3.10 Close the “EDIT JOB” spreadsheet and save the .job file.

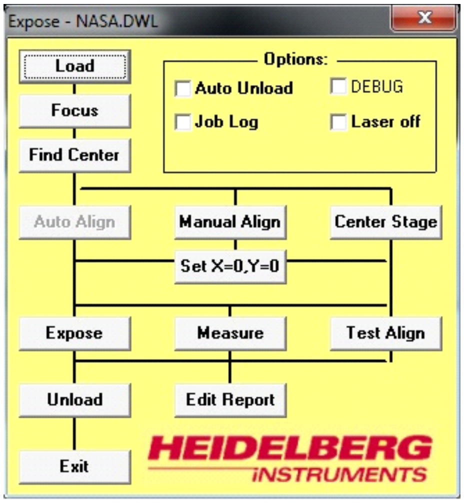
3.11 Turn on the laser.


Heidelberg Instruments DWL66
File Job Setup Measure Tools Service Laser Help
IF: OK
IF ?
IF R
Laser
ON
10mm WriteHead
binary exposure mode

Job File = PFM10MM.DWL Field Ali. = PFM10MM.FA
Map File = PFM10MM.MAP

3.12 Go to JOB – RUN JOB. The job exposure window tree will appear along with the field array map.

3.13 Click load. Wait for the stage to move out to the load position.

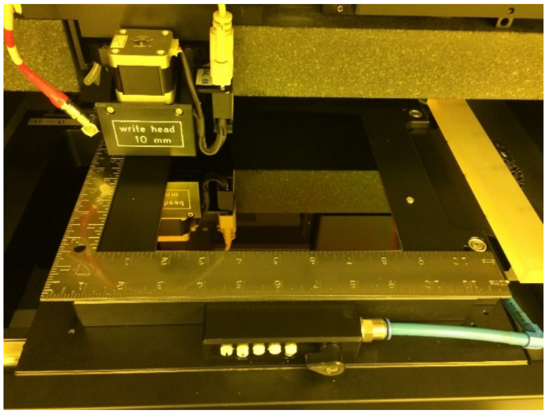


Options:

□ Auto Unload □ DEBUG

□ Job Log □ Laser off

3.14 Open the door and place the substrate in the middle of the stage. Turn on the vacuum.



3.15 Close the door.

3.16 Click OKAY and the state will move back to the center position.

3.17 Very that the write-head is right above the substrate (Caution!)

3.18 Click "focus". After a few seconds, the tool will find the correct focus.

3.19 Click "find center". The center window will appear. Click "start". The tool will find each edge of the substrate and then ask if you want to set the new origin. Click YES.

3.20 Click "expose". The system will begin with field one and expose the substrate. After a few minutes, it will display an estimated write time.

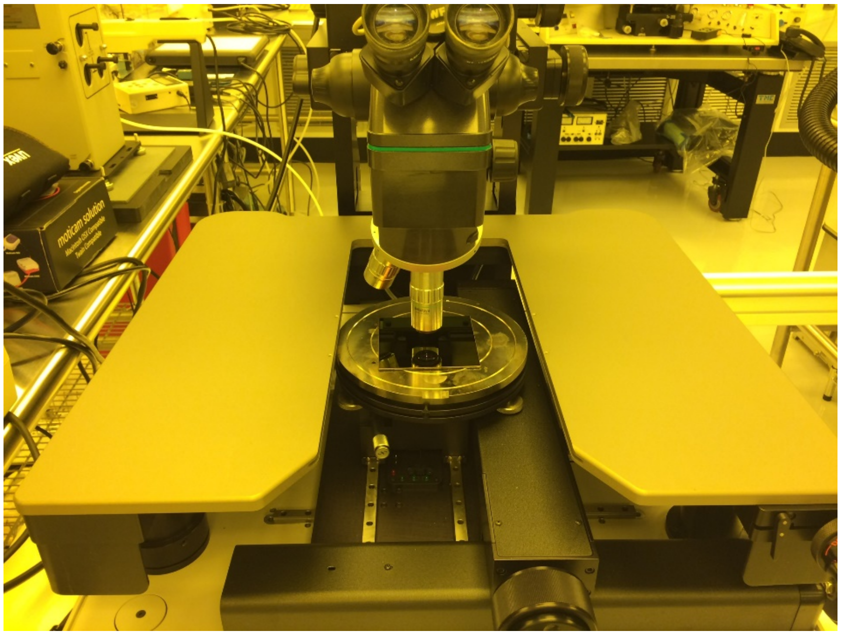
3.21 After the exposure is complete, click "unload".

3.22 Open the door, turn off the vacuum, and remove the substrate.

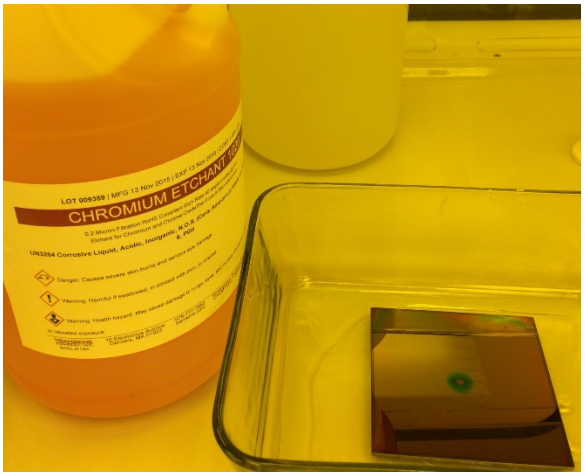
4. After the Lithography Process

4.1 Develop the PR in AZ400K developer for 60 seconds. Then, rinse and dry the sample.

4.2 Inspect the PR developed sample.

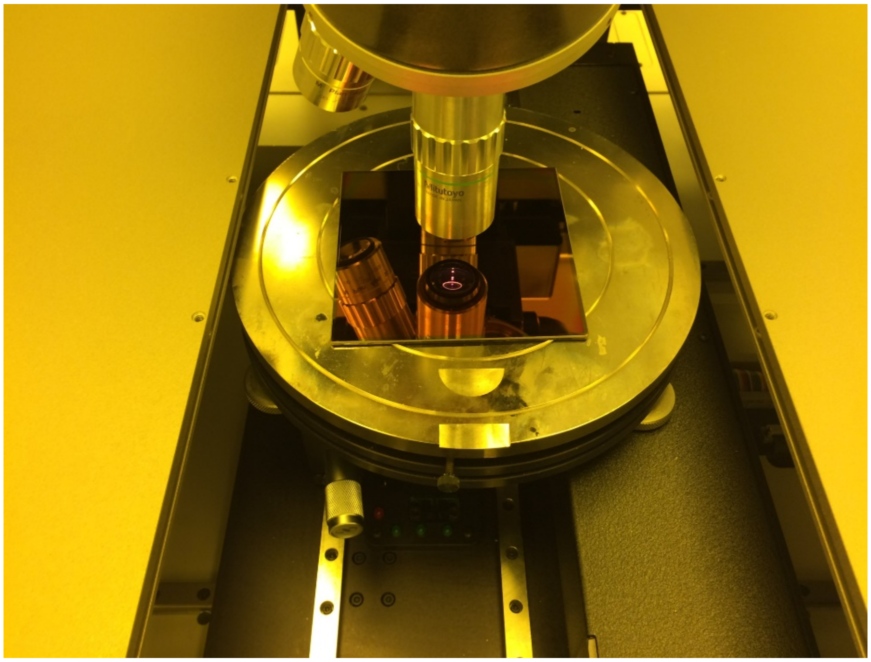


4.3 Etch in Cr etchant 1020 for 5 minutes at 65°C. Then, rinse and dry.

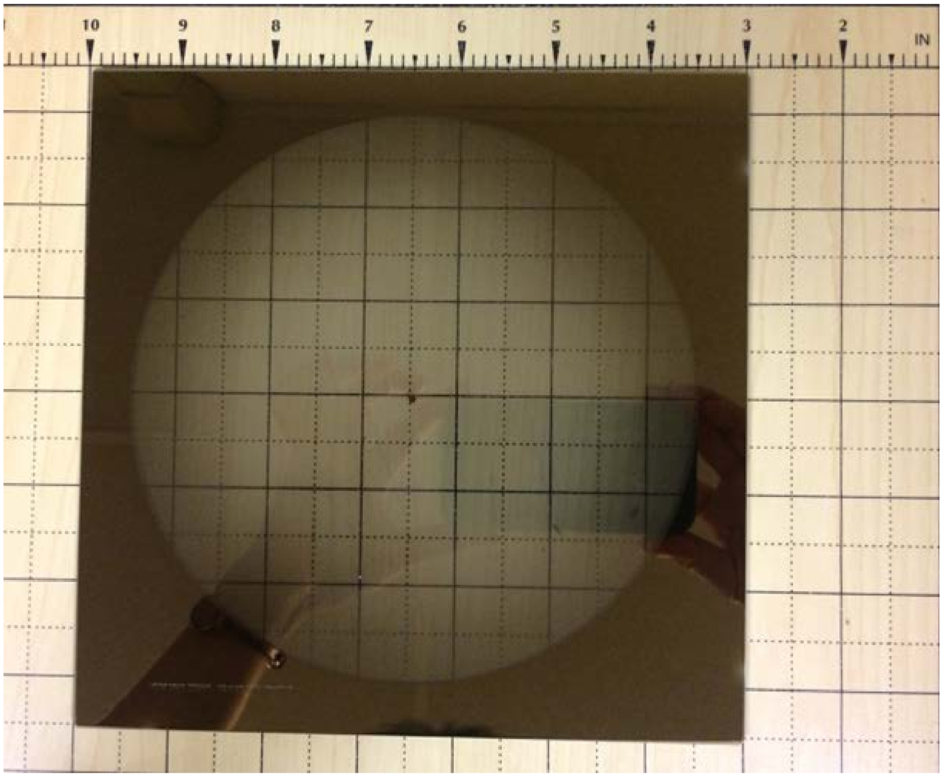


4.4 Remove the PR and clean the mask in Nanostrip for 5 minutes. Then, rinse and dry.

4.5 Conduct a final inspection using both the optical microscope and the Scanning Electron Microscope (SEM).



4.6 6 inch x 6 inch PS design on 7 inch x 7 inch wafer.



10
9
8
7
6
5
4
3
2
IN

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*
01/09/2017

2. REPORT TYPE
Technical Memorandum

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE

Rapid Generation of Large Dimension Photon Sieve Designs

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Hariharan, Shravan; Fitzpatrick, Sean; Kim, Hyun-Jung; Julian, Matthew; Sun, Wenbo; Tedjojuwono, Ken K.; MacDonnell, David G.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
432938.09.01.07.05.05

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Langley Research Center
Hampton, Virginia 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER

L-20860

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA-TM-2017-219664

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified
Subject Category 88
Availability: NASA STI Program (757) 864-9658

13. SUPPLEMENTARY NOTES

14. ABSTRACT
A photon sieve is a revolutionary optical instrument that provides high resolution imaging at a fraction of the weight of typical telescopes (areal density of 0.3 kg/m2 compared to 25 kg/m2 for the James Webb Space Telescope). The photon sieve is a variation of a Fresnel Zone Plate consisting of many small holes spread out in a ring-like pattern, which focuses light of a specific wavelength by diffraction. The team at NASA Langley Research Center has produced a variety of small photon sieves for testing. However, it is necessary to increase both the scale and rate of production, as a single sieve previously took multiple weeks to design and fabricate.

15. SUBJECT TERMS

Instrument; Photon; Sieve; Telescope

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			STI Help Desk(email help@sti.nasa.gov
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	68	(757) 864-9658