ISO/IEC JTC 1/SC 29/WG1 **N100xxx**

100th JPEG meeting – Covilhã, Portugal – July 17-21, 2023

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION

ISO/IEC JTC 1/SC 29/WG1
(ITU-T SG16)

Coding of Still Pictures

JBIG
Joint Bi-level Image Experts Group

JPEG
Joint Photographic Experts Group

TITLE:	Numerical Reconstruction Software for Holography (NRSH) User Guide
SOURCE:	Tobias Birnbaum, Roberto Corda, Antonin Gilles, Raees Kizhakkumkara Muhamad, Cristian Perra, Peter Schelkens
PROJECT:	JPEG Pleno Holography
VERSION:	13.0
STATUS:	For review
REQUESTED ACTION:	None
DISTRIBUTION:	WG1

Contact:
ISO/IEC JTC 1/SC 29/WG 1 Convener – Prof. Touradj Ebrahimi
EPFL/STI/IEL/GR-EB, Station 11, CH-1015 Lausanne, Switzerland
Tel: +41 21 693 2606, Fax: +41 21 693 7600, E-mail: Touradj.Ebrahimi@epfl.ch

TABLE OF CONTENTS

1. GENERAL USAGE

1.1 USER FUNCTIONS

The Numerical Reconstruction Software for Holography (NRSH) is a software written in MATLAB® based on [1], which allows for the reconstruction of b<>com, Interfere, EmergImg-HoloGrail and WUT Display holograms.

The tool comprises three main functions: *getSettings*, *nrsh* and *nrsh_print*. *getSettings* initializes reconstruction parameters, *nrsh* reconstructs medium resolution holograms (up to $16K \times 16K$), and *nrsh_print* reconstructs high-resolution holograms for printing applications (up to $200K \times 200K$).

To use this software, the following MATLAB® toolboxes must be installed and enabled:

- Image Processing Toolbox;
- Signal Processing Toolbox;
- Statistics and Machine Learning Toolbox.

A MATLAB® version equal to r2017b or newer is recommended.

1.2 RECONSTRUCTION PARAMETERS

In NRSH, most input reconstruction setting are read by nrsh and nrsh_print through a structure called *info*. This structure is initialized by function *getSettings*, whose declaration is given by:

info = getSettings(varargin)

This function takes as input a list of parameter name and value pairs, such that

info = getSettings('paramName1', paramValue1, 'paramName2', paramValue2, ...)

It returns structure *info* with fields *paramName1= paramValue1*, *paramName2= paramValue2*, etc.

Allowed parameters are the following:

- *usagemode* : (string or char array) – optional, default is 'exhaustive'
  Usage mode of nrsh. It can take four different values:
  - 'exhaustive': use combination of all possible viewpoints
  - 'individual': use individual viewpoints as listed
  - 'dynamic': use individual viewpoints as listed and save them as a video
  - 'complex': reconstruct the complex light field in the object plane, and disable non-invertible transforms (apertures, clipping, filters)
- *ap_sizes* : (numeric cell or array) – optional, default is empty
  Synthetic aperture size. If the synthetic aperture declaration is expressed as angles (cf. Section 2.2), it must be a single value expressed in degrees or a row or column vector of values. If the synthetic aperture declaration is expressed in pixels, it must be a 1 x n cell array, in which every element is a 1 x 2 vector that expresses the aperture size in pixels (height x width). If left empty, no aperture is applied.
- *h_pos* : (numeric array) – optional, default is 0
  If the synthetic aperture declaration is expressed as angles, it represents the horizontal angles, in degrees, at which the synthetic aperture will be placed. If the synthetic aperture declaration is expressed in pixels, it represents the horizontal position at which the synthetic aperture will be placed, expressed in

the range [-1, 1] where -1 is the leftmost position, while 1 is the rightmost position. In both cases (angle or pixel based) it can be a single value or a row or column vector of values.

- *v_pos* : (numeric array) – optional, default is 0
  If the synthetic aperture declaration is expressed as angles, it represents the vertical angles, in degrees, at which the synthetic aperture will be placed. If the synthetic aperture declaration is expressed in pixels, it represents the vertical position at which the synthetic aperture will be placed, expressed in the range [-1, 1] where -1 is the lowermost position, while 1 is the uppermost position. In both cases (angle or pixel based) it can be a single value or a row or column vector of values.
- *apertureinpxmode* :
  (boolean) – optional, default is true
  True to use pixel-based apertures, false for angle-based.
- *clip_min* : (numeric array) – optional, default is empty
  Minimal intensity values for clipping. It can be a single value or a row or column vector of values (one per reconstruction). If left empty, a percentile clipping may be performed (cf. *perc_clip* in Section 1.3).
- *clip_max* : (numeric array) – optional, default is empty
  Maximal intensity values for clipping. It can be a single value or a row or column vector of values (one per reconstruction). If left empty, a percentile clipping may be performed (cf. *perc_clip* in Section 1.3).
- *use_first_frame_reference* :
  (boolean) – optional, default is true
  True to use the computed absolute clipping values of the first reconstruction for the next ones, false otherwise.
- *dataset* : (string or char array) – optional, default is empty
  Dataset to which the hologram (hol) belongs. It should be one of the following character arrays:
  - 'bcom8'
  - 'bcom32'
  - 'bcom32_bin'
  - 'interfere'
  - 'interfere_bin'
  - 'interfere4'
  - 'interfere4_bin'
  - 'emergimg'
  - 'emergimg_bin'
  - 'wut_disp'
  - 'wut_disp_on_axis'
  - 'wut_disp_on_axis_bin'
- *cfg_file* : (string or char array) – optional, default is empty
  Path to configuration file. It should be a character vector. See Section 1.3 for more details about configuration files.
- *name_prefix* : (string or char array) – optional, default is empty
  Name prefix for the files written.
- *outfolderpath* : (string or char array) – optional, default is './figures'
  Path to the output folder for figures.

- *direction* (string or char array) – optional, default is 'forward'
  Propagation direction. It has effect only if *usagemode* = 'complex', and should take one of the following values:
  - 'forward': forward transform (propagation towards the object plane)
  - 'inverse': inverse transform (propagation towards the hologram plane)
- *resize_fun* : (string, char array or function handle) – optional, default is empty
  Resize/clipping/down-sampling function handle to use on reconstructions. If 'DR' is provided, diffraction-limited reconstruction is performed using phase-space bandwidth limitation to reduce the resolution of reconstructed image (cf. document wg1m89038). If left empty, no resizing is performed.
- *targetres* : (numeric array) – optional, default is empty
  Target resolution of the final video, when using *resize_fun* = 'DR'. No frame will have higher resolution. A single aperture size will be calculated for all frames. If left empty and *resize_fun* = 'DR', the diffraction-limited reconstruction will be based on the input aperture size *ap_sizes*.
- *fps* : (scalar number) – optional, default is 10
  Frame rate of final video. It has effect only if *usagemode* = 'dynamic'.

Additionally, *getSettings* accepts any configuration parameter listed Section 1.3. In that case, the parameter values given to *getSettings* overwrite those defined in configuration files.

1.3 CONFIGURATION FILES

Some hologram's reconstruction parameters can also be provided to the software through simple text configuration files. In order to use a specific configuration file, the user has to declare the file's path through the *info.cfg_file* field. Only one configuration file can be used per function call.

During the reading process, a line is ignored if it is empty or if it starts with "#". The parameter and its value(s) should be separated by ":" and every row should contain a single parameter-value(s) pair. The *config_files* folder contains some examples of pre-built configuration files. The allowed parameters are:

- *wlen* : (numeric array) – mandatory
  Wavelength(s) for the reconstruction (in meters). It is a single value if the hologram is monochrome; it is composed of 3 values (R, G and B wavelengths, in this order) separated by a comma if the hologram is RGB.
- *pixel_pitch* : (scalar number) – mandatory
  Pixel pitch of the input hologram (in meters). It should be a single value.
- *method* : (string or char array) – mandatory
  Reconstruction method. The allowed values are:
  - 'ASM': Angular Spectrum propagation method
  - 'Fresnel': Fresnel diffraction method
  - 'Fourier-Fresnel': for WUT Display holograms
- *zero_pad* : (boolean) – optional, default is false
  In configuration files, it should be a binary value: 1 to zero-padding the hologram before the reconstruction, 0 otherwise. It has effect only if the reconstruction method is ASM.
- *apod* : (boolean) – optional, default is false
  In configuration files, it should be a binary value: 1 to apodize the synthetic aperture using a bidimensional Hanning window, 0 otherwise. The apodization is not performed if no synthetic aperture is declared.
- *perc_clip* : (boolean) – optional, default is false

In configuration files, it should be a binary value: 1 to perform a percentile clipping, in order to remove high amplitude spikes, 0 otherwise.

- *perc_value* : (scalar number) – optional, default is 100
  Percentile threshold for the clipping procedure. It is a scalar between 0 and 100. e.g. if *perc_value* = 95 all the amplitudes higher than this percentile are set to the value corresponding to *perc_value* = 95. It has effect only if *perc_clip* = 1.
- *hist_stretch* : (boolean) – optional, default is false
  In configuration files, it should be a binary value: 1 to perform a linear stretching in the range [0,1] of the complex magnitude values obtained after the reconstruction, 0 otherwise.
- *save_intensity* : (boolean) – optional, default is false
  In configuration files, it should be a binary value: 1 to save the intensity of complex wave field as MAT and/or PNG file, 0 to save its amplitude.
- *save_as_mat* : (boolean) – optional, default is false
  In configuration files, it should be a binary value: 1 to save the reconstructed image or complex wave field as MAT file, 0 otherwise. If *perc_clip* and/or *hist_stretch* has been activated, the saved data include also these enhancement steps. The filename follows the structure: *ConfigurationFileName_Hpos_Vpos_ApSize_RecDist.mat* and is saved in the *./outfolderpath/ConfigurationFileName/* path.
- *save_as_image* :
  (boolean) – optional, default is true
  In configuration files, it should be a binary value: 1 to save the reconstructed image as PNG, with a bit depth specified by the parameter *bit_depth*, 0 otherwise. The filename follows the structure: *ConfigurationFileName_Hpos_Vpos_ApSize_RecDist.png* and is saved in the *./outfolderpath/ConfigurationFileName/* path.
- *show* : (boolean) – optional, default is false
  In configuration files, it should be a binary value: 1 to show in a MATLAB® figure the reconstructed image, 0 otherwise. Every reconstruction is showed in a different figure.
- *bit_depth* : (scalar number) – optional, default is 8
  Bit depth of the reconstructed image *hol_image* returned by NRSH and bit depth of the PNG image(s) saved to disk (if *save_as_image* =1). It should be only 8 or 16.

The following parameters are related to binary holograms only

- *reffronorm* : (numeric array) – optional, default is 1
  Frobenius norm per color channel of complex valued ground truth, only required for binary DHs. It can be arbitrary ($\neq 0$) for binary DHs not obtained from complex-valued pendants.
- *offaxisfilter* : (string or char array) – optional, default is 'h'
  Off-axis filter type. Should be 'h', or 'v'. If 'v', nrsh simulates an incident off-axis planar illumination above the hologram with vertical fringes. If 'h', nrsh simulates an incident off-axis planar illumination above the hologram with horizontal fringes.

The following parameter is related to Interfere IV and WUT Display reconstructions only:

- *ref_wave_rad*: (scalar number) – mandatory if *dataset* = 'wut_disp', 'wut_disp_on_axis', 'wut_disp_on_axis_bin', 'interfere4' or 'interfere4_bin'
  Reference wave radius, in meters. It should be a single value.

The following parameters are related to WUT Display reconstructions only:

- *DC_filter_type*: (string or char array) – optional, default is 'wut'
  Filter used to suppress the DC component. It should be a character vector. The value of this parameter can be one of the following:
  - wut Filter included in WG1M83031 [5]
  - bartlett Bartlett window.
  - barthannwin Modified Bartlett-Hanning window.
  - blackman Blackman window.
  - blackmanharris Minimum 4-term Blackman-Harris window.
  - bohmanwin Bohman window.
  - chebwin Chebyshev window.
  - flattopwin Flat Top window.
  - gausswin Gaussian window.
  - hamming Hamming window.
  - hann Hann window.
  - kaiser Kaiser window.
  - nuttallwin Nuttall defined minimum 4-term Blackman-Harris window.
  - parzenwin Parzen (de la Valle-Poussin) window.
  - rectwin Rectangular window.
  - taylorwin Taylor window.
  - tukeywin Tukey window.
  - triang Triangular window
- *DC_filter_size*: (scalar number) – optional, default is 0.5
  Size of the DC filter, expressed as fraction of the output image size. The value should be in the range (0,1).
- *img_flt* : (scalar char) – optional, default is empty
  Allows filtering the orthoscopic or pseudoscopic image in the reconstruction. 'R' filters the right part of the reconstruction, 'L' the left one. If left empty, no image filtering process is performed.
- *shift_yx_R* : (numeric array) – optional, default is 0
  Shift values for the R channel, in meters. It should be a pair of values, separated by a comma.
- *shift_yx_G* : (numeric array) – optional, default is 0
  Shift values for the G channel, in meters. It should be a pair of values, separated by a comma.
- *shift_yx_B* : (numeric array) – optional, default is 0
  Shift values for the B channel, in meters. It should be a pair of values, separated by a comma.

The following parameters are related to high-resolution holograms only, to be reconstructed with nrsh_print:

- *hologramName*: (string or char array) – mandatory if *dataset* = 'bcom_print' or 'etri_print'
  Name of the hologram. It should be a character vector.

- *format*: (string or char array) – mandatory if *dataset* = 'bcom_print' or 'etri_print'
  Format of the hologram. It should be a character vector. The value of this parameter can be one of the following:
  - complex: Complex-valued hologram, stored in amplitude and phase images
  - bilevel: Bilevel-phase hologram
- *segmentsNum*: (numeric array) – mandatory if *dataset* = 'bcom_print' or 'etri_print'
  Number of holographic segments. It should be a pair of values (height, width, in this order), separated by a comma.
- *segmentsRes*: (numeric array) – mandatory if *dataset* = 'bcom_print' or 'etri_print'
  Resolution of holographic segments. It should be a pair of values (height, width, in this order), separated by a comma.
- *subSegmentsRes*: (numeric array) – optional, default is *segmentsRes*
  Resolution of holographic sub-segments for reconstruction [3]. It should be a pair of values (height, width, in this order), separated by a comma
- *spectrumScale*: (scalar number) – optional, default is 1
  Scale for the FFT calculations [3]. It should be a single integer value.

1.4 RENDERING OF MEDIUM-RESOLUTION HOLOGRAMS USING NRSH

The *nrsh* declaration is the following:

[hol_rendered, clip_min, clip_max] = nrsh(hol, rec_dists, info, varargin)

The input parameters are:

- *hol*: (string, char array or numeric array) – mandatory
  Hologram to reconstruct. It can be a matrix that has been previously loaded in workspace or a path to a folder (provided as character array) that contains the files that represent the hologram. If left empty, the hologram is loaded manually through the GUI (cf. Section 2.1).
- *rec_dists*: (numeric array) – mandatory
  Reconstruction distances in meters. It can be a single value, or a row or column vector of values.
- *info*: (struct) – mandatory
  Reconstruction parameters, initialized with *getSettings* (cf. Section 1.2)

The user can also overwrite the parameters defined in configuration files or in structure *info* by passing them in this specific order as additional input parameters through *varargin: usagemode, ap_sizes, h_pos, v_pos, clip_min, clip_max.*

The output parameters are:

- *hol_rendered*: (numeric array)
  Reconstruction of the input hologram, returned as standard unsigned integer image (8 or 16 bpp). Note that in case of multiple reconstructions, *hol_rendered* is the last reconstruction performed.
- *clip_min*: (numeric array)
  Minimal intensity of the numerical reconstructions. In case of multiple reconstructions, one value per reconstruction is returned.
- *clip_max*: (numeric array)
  Maximal intensity of the numerical reconstructions. In case of multiple reconstructions, one value per reconstruction is returned.

The behavior of nrsh depends on *usagemode*:

- If *usagemode* = 'exhaustive', the software calculates all possible combinations of *rec_dists, ap_sizes, h_pos, v_pos* and performs a reconstruction for each combination of values;
- If *usagemode* = 'individual', it performs a reconstruction for each individual viewpoint listed as input;
- If *usagemode* = 'dynamic', it performs a reconstruction for each individual viewpoint listed as input and saves them as a pseudo-video sequence for subjective testing;
- If *usagemode* = 'complex', it skips all non-invertible transforms (apertures, filters, clipping and resizing) to obtain the complex-valued wavefield in the object plane (used for object-plane coding).

In the angle-based case, the synthetic aperture size and position are automatically determined from the three angles (*dof_angles, h_angles, v_angles*) as recommended in WG1M82039 [2]. See Section 2.2 for more details.

The software automatically resizes, crops or uses any "resize" function supplied as *resize_fun* argument to avoid writing large size images to the disk. Resizing is performed on the real data, before any conversion to integers. Clipping is done by default with a percentile value for the first reconstruction, and then using the same absolute threshold for the next ones. It is also possible to provide absolute clipping values *clip_min* and *clip_max* per reconstruction.

The reconstructions can be saved as MAT files and/or as PNG images (8 or 16 bpp) and are stored in the *./outfolderpath/ConfigurationFileName/* path. The file names are structured as follows:

<name_prefix><ConfigurationFileName>_<Hpos>_<Vpos>_<ApSize>_<RecDist>.(mat | png)

Finally, if *usagemode* = 'dynamic' it creates a video from these images called

<name_prefix><ConfigurationFileName>_nFramesXXXatYYYFPS.mp4, where *XXX* is the total number of frames N and *YYY* is the number of frames per second given by parameter *fps*.

1.5 RENDERING OF HIGH-RESOLUTION HOLOGRAMS FOR PRINTING APPLICATIONS

Function *nrsh_print* reconstructs high-resolution holograms for printing applications (up to 200K×200K). Its declaration is the following:

[hol_rendered, clip_min, clip_max] = nrsh_print(hol, info, viewpoints, varargin)

The input parameters are the same as *nrsh*, except for:

- *hol* : (string or char array) – mandatory
  Folder path in which the hologram segments are located. It should be a character vector.
- *viewpoints* : (numeric array) – mandatory
  Viewpoints coordinates, expressed in normalized coordinates in the range ([-1, 1], [-1, 1], [0, inf]), where -1 is the lowermost or leftmost position, while 1 is the uppermost or rightmost position. It should be a $(n \times 3)$ array, where each row corresponds to a single viewpoint.

The user can also overwrite the parameters defined in configuration files or in structure *info* by passing them in this specific order as additional input parameters through *varargin: hologramname, format, segmentsnum, segmentsres, subsegmentsres, spectrumscale, clip_min, clip_max*.

The output parameters are the same as *nrsh*, except for:

- *hol_rendered* : (numeric array)
  Reconstructions of the input hologram, returned as a four dimensional array of standard unsigned integer images (8 or 16 bpp), with a size of [height, width, channels, n].

Due to their very large resolution, the high-resolution holograms available in the b<>com and ETRI holographic datasets are divided into *segmentNum* rectangular segments of resolution *segmentRes* to facilitate their processing. These segments are stored into bitmap or tiff files as described in [3, 4].

The software reconstructs the hologram by extracting coefficients corresponding to light rays reaching the viewers' pupils from its space-frequency distribution, as described in [3]. For the sake of simplicity, the space-frequency distribution of the hologram is computed by dividing segments into sub-segments of resolution *subSegmentRes* and computing the 2D Fourier Transform of each sub-segment.

The reconstructions can be saved as MAT files and/or as PNG images (8 or 16 bpp) and they will be stored in the *<outFolderPath>/<ConfigurationFileName>/* path. The file names are structured as follows:

<name_prefix><ConfigurationFileName>_<viewpoint>.(mat | png)

2. ADDITIONAL INFORMATION

2.1 HOLOGRAM LOADING WITH NRSH

2.1.1 MANUAL LOADING USING THE GUI

When *nrsh* is launched with *hol* left empty, a window will open and the user can select the dataset to which the hologram belongs (Fig. 1).

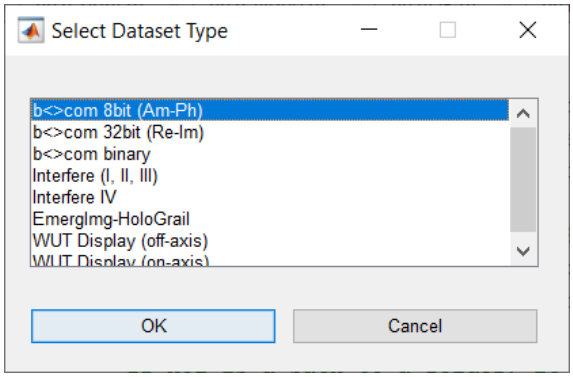


Figure 1: dataset selection

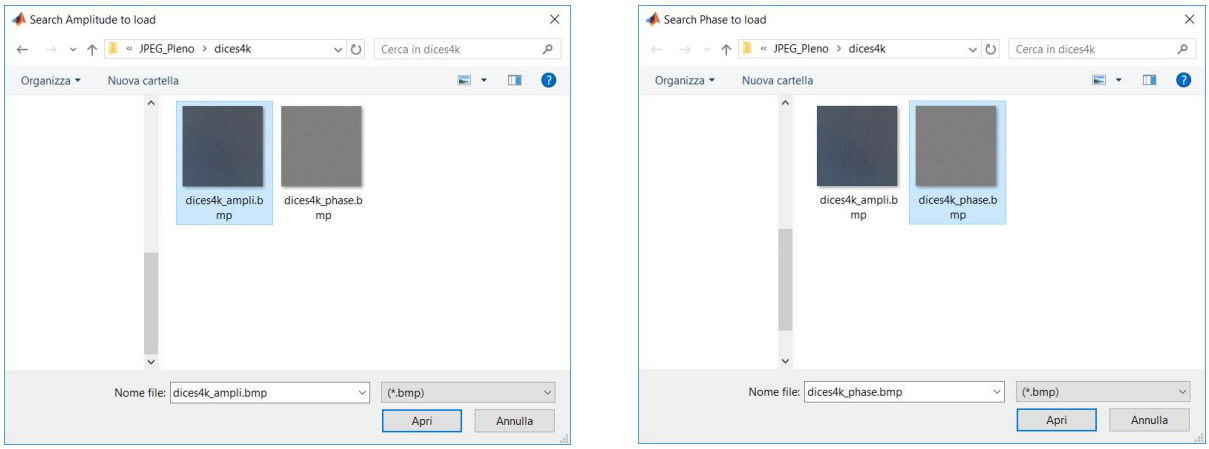


Figure 2: b<>com dataset file selection selection

When the user selects “b<>com 8bit”, two dialog windows will be opened: the first one asking for the amplitude bmp file and the second one asking for the phase bmp file (Fig.2).

If the user selects “b<>com 32bit”, the file selection sequence is the same as the 8 bit case. The only difference is that the files are not bmp but exr.

When the user selects “b<>com binary”, “Interfere (I, II, III)”, “Interfere IV”, “EmergImg-HoloGrail” or “WUT Display (on-axis)”, a dialog window will be opened, where the mat file corresponding to the hologram can be selected (Fig. 3).

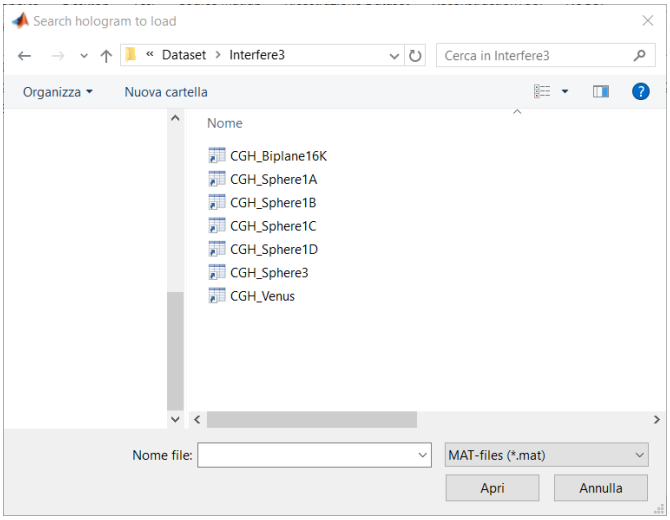


Figure 3: Interfere or EmergImg-HoloGrail file selection

If the user selects “WUT Display (off-axis)”, the first dialog window allows to select the R channel (in case of RGB holograms) or the monochrome hologram (Fig. 4, left). After the file selection, a new dialog window will appear (Fig. 4, right). If the hologram is RGB, the user responds “No” and the loading of G and B channels is allowed. If the hologram is monochrome, the user has to respond “Yes” to terminate the loading process.

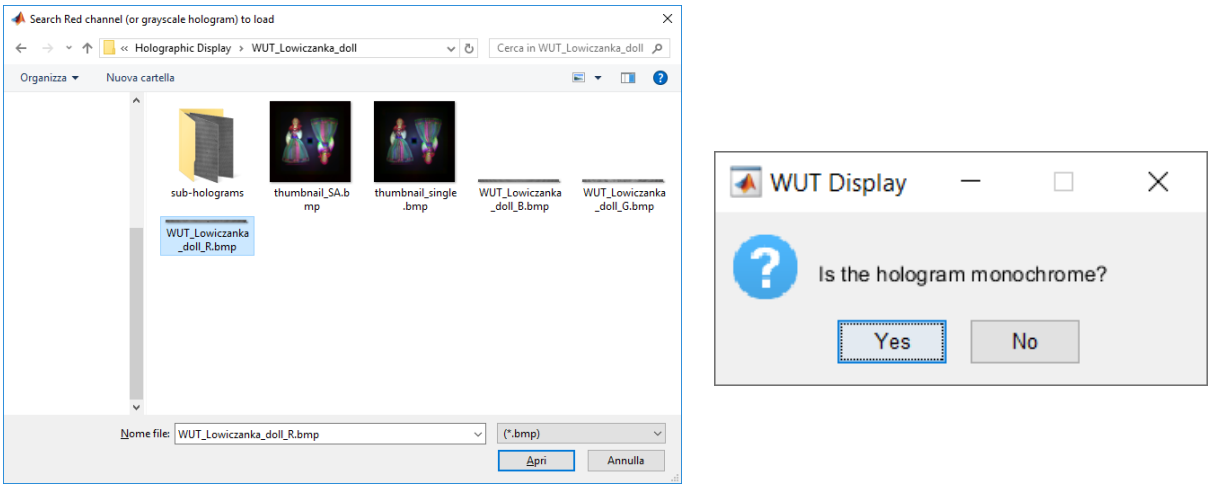


Figure 4: WUT display file selection

2.1.2 AUTOMATIC IDENTIFICATION AND LOADING FROM FOLDER

When *nrsh* is launched with *hol* being a folder path, it automatically loads the hologram from the corresponding folder. The hologram should belong to one of the supported datasets, listed in Section 1.2. In order to successfully identify a hologram, the folder in which the hologram is stored should not contain any other file other than those that constitute the hologram to load, but it can contain other folders. The holograms are identified by their filenames, thus is strongly recommended to use the original filenames.

As an example, the reconstruction of b<>com's Dices8K with the automatic hologram loading is showed. In the root of NRSH, a folder named "example" has been manually created, that contains the files that represent the hologram (Fig. 5).

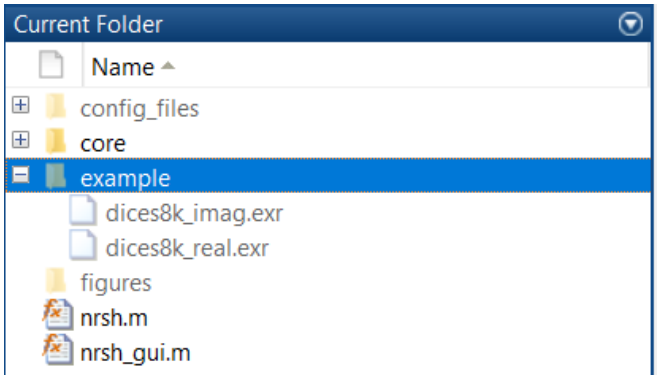


Figure 5: Automatic hologram loading example

2.1.3 MANUAL IDENTIFICATION AND LOADING FROM A VARIABLE

The *nrsh* function can start a reconstruction from a hologram that has been previously loaded as a variable in the Workspace. In this case, the variable that represents the hologram is passed as first parameter *hol*, while its dataset should be identified in *info.dataset*.

As an example, the reconstruction of EmergImg-Holograil's v2 Astronaut is showed. The hologram has been already loaded in the Workspace (Fig. 6), and the variable that represents the hologram is u1.

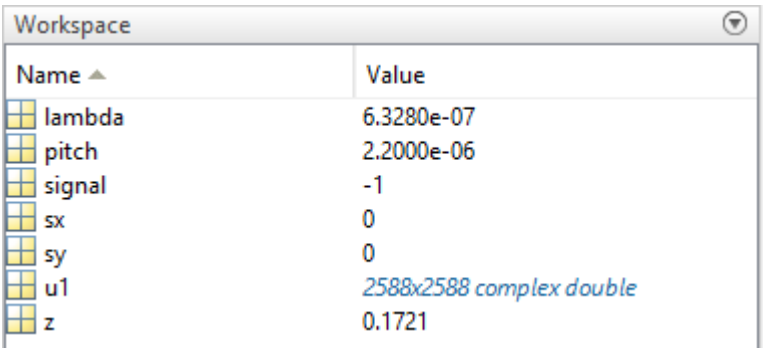


Figure 6: Loading from a variable example

2.2 SYNTHETIC APERTURE

This version of NRSH supports two different conventions to declare the position and the size of the synthetic aperture: the pixel-based convention (*apertureinpxmode*=true) and the angle-based convention (*apertureinpxmode*=false).

In the pixel-based convention, *ap_sizes* must be a 1 x n cell array, with every element being a 1 x 2 vector that expresses the height and width (in this order) of the synthetic aperture in pixels. *h_pos* and *v_pos* are values in the [-1, 1] range that represent the horizontal and vertical positions of the synthetic aperture.

On the contrary, in the angle-based convention, *ap_sizes*, *h_pos* and *v_pos* must be single numeric values or row vectors of values, representing angles expressed in degrees.

2.2.1 PIXEL-BASED SYNTHETIC APERTURE

In pixel-based mode, the position of the synthetic aperture is set with *h_pos* and *v_pos* , with values in the range [-1, 1], similarly to WUT FDHRender software [5].

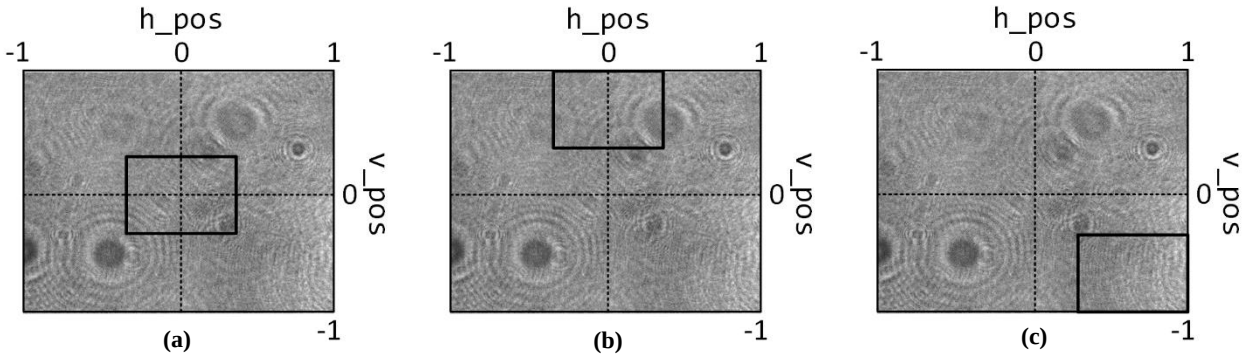


Figure 7: Synthetic aperture positions (pixel based)
(a) h_pos=0, v_pos=0, (b) h_pos=0, v_pos=1, (c) h_pos=1, v_pos=-1

Fig. 7 shows some examples of positioning. With this convention, rectangular apertures can also be declared. The central position in the hologram plane is given by *h_pos=0, v_pos=0* (Fig. 7a). A positive value of *v_pos* shifts the aperture vertically upwards (Fig. 7b), while a positive value of *h_pos* shifts the aperture horizontally to the right (Fig. 7c).

In the pixel-based mode, if one dimension of the synthetic aperture is set to zero, the hologram is reconstructed without synthetic aperture (full aperture): e.g. if *ap_sizes={[0,2048]}*, no synthetic aperture will be used. However, if there is the need to combine reconstructions with full aperture and reconstructions with a synthetic aperture, it is suggested to set both dimensions to zero to declare the full aperture case (e.g. *ap_sizes={[0, 0], [2048, 2048]}*).

2.2.2 ANGLE-BASED SYNTHETIC APERTURE

The angle-based convention follows the recommendations included in [2] for the automatic selection of the synthetic aperture position and size. Two angles are used to determine the aperture position in the hologram plane: the horizontal angle θ = arctan(xc/zrec) and vertical angle φ = arctan(yc/zrec). As shown in Fig. 8, xc and yc are the horizontal and vertical distances respectively, between the synthetic aperture center and the hologram plane center, while zrec is the reconstruction distance.

In this case, angles θ and φ are represented by the *h_pos* and *v_pos* parameters, respectively. The synthetic aperture size is determined with a single angle, ψ (Fig. 8, right), represented by the *ap_size* parameter.

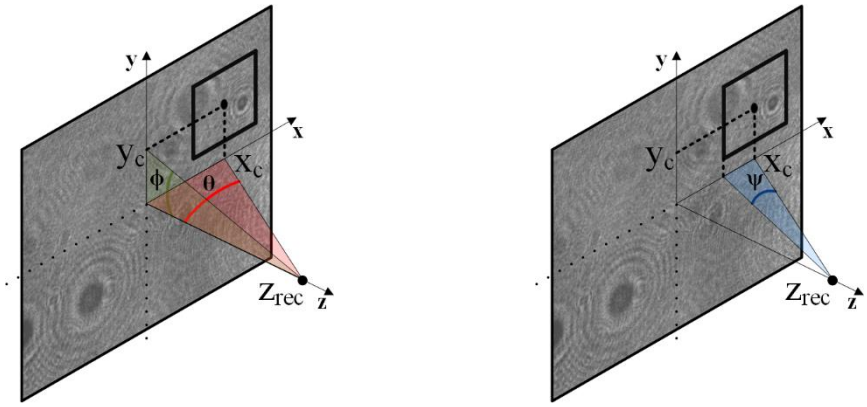
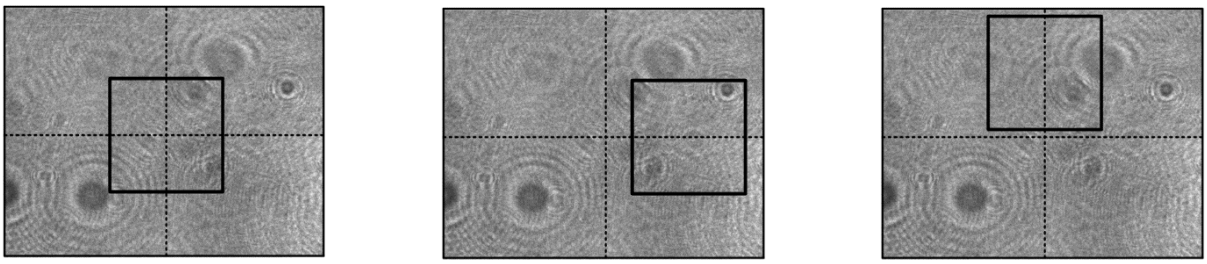


Figure 8: Synthetic aperture position expressed with θ and φ (left) and size expressed with ψ (right).

When *h_pos=0* and *v_pos=0* the synthetic aperture is located in the center of the hologram plane (Fig. 9a). A positive value of *h_pos* shifts the synthetic aperture horizontally to the right, with respect to the center of the hologram plane. On the contrary, a negative value of *h_pos* shifts the synthetic aperture horizontally to the left. A positive value of *v_pos* shifts the synthetic aperture vertically upwards, while a negative value of *v_pos* shifts the synthetic aperture vertically downwards.



(a) **(b)** **(c)**

Figure 9: Synthetic aperture positions (angle-based):
(a) h_pos=0, v_pos=0, (b) h_pos=positive_value, v_pos=0, (c) h_pos=0, v_pos=positive_value

Finally, when *ap_sizes=0* the hologram is reconstructed without synthetic aperture (full aperture). Please refer to [2] for further information.

2.3 NRSH PROMPT MESSAGES AND AVERAGE TIMES CALCULATION

During execution, the software provides information about the input parameters used for the reconstructions (Fig. 10). If the user provides different combinations of *h_pos*, *v_pos* and *ap_sizes* (which involve different synthetic aperture positions and/or sizes), all these combinations will be validated before starting the reconstructions. The validation process checks for incorrect (in size or position) synthetic aperture declarations.

If incorrect combinations are detected:

1. If all combinations are incorrect, the execution is aborted (Fig. 11).
2. If only some combinations are wrong, the software continues the execution ignoring the wrong combinations (Fig. 12).

```
*****************************************************************************************
*************************Configuration setup:************************
*****************************************************************************************
                        rec_dists : 0.3964
                         cfg_file : config_files\interfereIV\deepchess2_000.txt
                             wlen : 5.32e-07
                       pixel_pitch : 3.45e-06
                           method : Fourier-Fresnel
                         zero_pad : 0
                             apod : 0
                        perc_clip : 1
                       perc_value : 99.95
                     hist_stretch : 1
                     ref_wave_rad : 0.9986
                      save_as_mat : 0
                             show : 0
                    save_as_image : 1
                        bit_depth : 16
                   save_intensity : 0
                        usagemode : exhaustive
                         ap_sizes : 2048  2048
                            h_pos : -0.5          0         0.5
                            v_pos : 0
                          dataset : interfere4
                       resize_fun :
                        targetres : 2048  16384
                         clip_min : -1
                         clip_max : -1
          use_first_frame_reference : 1
                        direction : forward
                      name_prefix :
                     outfolderpath : ./figures
                  apertureinpxmode : 1
                        reffronorm : 1
                               fps : 10
```

Figure 10: Information about reconstruction parameters

```
Synthetic aperture out-of-bound check...
Warning: The following combination(s) of h_angle/v_angle/dof_angle/rec_dist generate(s) an out-of-bound synthetic aperture:
> In aperture_angle_checker (line 148)
  In nrsh (line 129)
h_angle=0, v_angle=0, dof_angle=40, rec_dist=0.0023
Error using aperture_angle_checker (line 158)
There are no other valid combinations! Execution aborted.

Error in nrsh (line 129)
    rec_par_idx=aperture_angle_checker(size(hol,1), size(hol,2),rec_par_idx,...
```

Figure 11: synthetic aperture check: no valid combination found

```
Synthetic aperture out-of-bound check...
Warning: The following combination(s) of h_angle/v_angle/dof_angle/rec_dist generate(s) an out-of-bound synthetic aperture:
> In aperture_angle_checker (line 148)
  In nrsh (line 129)
h_angle=0, v_angle=0, dof_angle=40, rec_dist=0.0023
```

Figure 12: synthetic aperture check: some valid combinations found

Finally, the execution time is shown. It includes following operations:

- synthetic aperture setup
- numerical propagation
- enhancement processes (as percentile clipping or histogram stretching)
- other dataset-dependent final operations

The following operations are not taken into account for execution time recording:

- user input/configuration file parameters loading
- user input check
- hologram loading process
- reconstruction visualization/saving processes

At the end of all reconstructions, the software provides the average time for one reconstruction (calculated as the arithmetic mean of all reconstructions performed), as shown in Fig.13.

```
Calculating theta=0 phi=0 psi=0 z=0.2955 ... done in 1.47 seconds.

Calculating theta=0 phi=0 psi=0 z=0.2965 ... done in 1.28 seconds.

Calculating theta=0 phi=0 psi=0 z=0.2975 ... done in 1.22 seconds.

Calculating theta=0 phi=0 psi=0 z=0.2985 ... done in 1.27 seconds.

Calculating theta=0 phi=0 psi=0 z=0.2995 ... done in 1.30 seconds.

Calculating theta=0 phi=0 psi=0 z=0.3005 ... done in 1.17 seconds.

Calculating theta=0 phi=0 psi=0 z=0.3015 ... done in 1.34 seconds.

Calculating theta=0 phi=0 psi=0 z=0.3025 ... done in 1.31 seconds.

Calculating theta=0 phi=0 psi=0 z=0.3035 ... done in 1.30 seconds.

Calculating theta=0 phi=0 psi=0 z=0.3045 ... done in 1.32 seconds.

Average reconstruction time (10 reconstruction(s)): 1.30 seconds.
>>
```

Figure 13: Reconstruction time information

2.4 WUT DISPLAY RECONSTRUCTION SUPPORT

In this version of nrsh, the reconstruction process of WUT Display holograms is derived from the FDHrender software, provided by WUT in WG1M83031 [5], adapting it as far as possible to the nrsh features and characteristics. Because it is not a 1:1 porting of WUT FDHrender software, the reconstructions results between the two software might be different. Moreover, some features of WUT FDHrender are not available in this version of nrsh (e.g. saving pixel pitch and physical size of the hologram after the reconstruction process).

The following functions, originally developed and provided by WUT in WG1M83031 [5], are employed and included in this version of NRSH without modification with respect to their original implementations:

- DCfilter.m
- subarray_yx.m

The following function, originally developed and provided by WUT in WG1M83031 [5], is employed and included in this version of NRSH with a minor modification with respect to its original implementation, that has been clearly indicated in the header and in the code of the function:

- saturate_gray.m

The above-mentioned functions are stored in the ./core/WUT_lib path of NRSH, where ./ is the root folder of NRSH.

3. VERSION HISTORY

Version 13.0 (wg1n100xxx):

- Contributors: Antonin Gilles (b<>com), Tobias Birnbaum (VUB/IMEC)
- Changes:
  - Bug fix for non-square resolutions in rgb_align and corrections to the user manual

Version 12.0 (wg1n100532):

- Contributors: Tobias Birnbaum (VUB/IMEC)
- Changes:
  - Closing bug: Missing isLast in argument list of rec_fresnel

Version 11.0 (wg1n100417):

- Contributors: Tobias Birnbaum (VUB/IMEC)
- Changes:
  - Added memory saving options, peak memory usage optimizations

Version 10.0 (wg1n100342):

- Contributors: Tobias Birnbaum (VUB/IMEC), Antonin Gilles (b<>com)
- Changes:
  - Merged rec_fresnel and rec_fourier_fresnel propagation scripts
  - Added rec_fresnel_deprecated for EmergImg datasets
  - Added commentary into getSettings
  - Generalized on-/off-axis check
  - Added orthoscopic reconstructions
  - Added verbosity, isBinary and isFourier flags into info struct
  - Improved print_setup
  - Improved defaultSettings
  - Improved validateSettings
  - Various bug fixes

Version 9.0 (wg1n100273):

- Contributors: Tobias Birnbaum (VUB/IMEC), Antonin Gilles (b<>com)
- Changes:
  - Added support for non-square pixel pitch in rec_asm
  - Bug fix with respect to persistent kernel in rec_asm
  - Added bandwidth limitation and zero-padding support in rec_fresnel
  - Added bandwidth limitation and zero-padding support in rec_fourier_fresnel

Version 8.0 (wg1n100197):

- Contributors: Tobias Birnbaum (VUB/IMEC), Antonin Gilles (b<>com)
- Changes:
  - Fixed nrshDR2single interface wrt. subj test pipeline
  - Renamed biplane config files to avoid overwriting
  - Bug fix: corrected num_rec behavior for z=0, e.g. seen for Breakdancers8k4k
  - Lower RAM usage if no return value is required
  - Added piano16kr config
  - WUT Mermaid bugfix (wg1m95002); Deprecated: saturate_gray (wg1m95003)

Version 7.0 (wg1n92044):

- Contributors: Antonin Gilles (b<>com), Tobias Birnbaum (VUB/IMEC)
- Changes:
  - New interface for nrsh, based on a configuration structure (info)
  - New interface for nrsh_print, based on a configuration structure (info)
  - Removed *nrsh_complex, nrsh_gui, nrsh_video, nrshDR, nrshDR2* and *nrshDR2single* (now included in *nrsh*)
  - Added support for Octave
  - Major updates to the user guide and minor bug fixes

Version 6.0 (wg1n91060):

- Contributors: Antonin Gilles (b<>com), Tobias Birnbaum (VUB/IMEC)
- Changes:
  - Added optional parameter *refFroNorm* to *nrsh_print* and *nrshDR2*
  - Added optional parameter *name_prefix* to *nrsh_complex, nrsh_gui, nrsh_video, nrshDR, nrshDR2* and *nrshDR2single*
  - Added optional specification of output dir in *nrsh_complex, nrsh_gui, nrsh_print, nrsh_video, nrshDR, nrshDR2* and *nrshDR2single* (may be required for disk-space reasons)
  - Added optional *doDynamic* flag in *nrsh_gui, nrshDR* and *nrshDR2*
  - Updated JPEG Pleno database file
  - Added binary holograms reconstruction scripts
  - Added new configuration files
  - Minor modifications and fixes to the user guide

Version 5.0 (wg1n90060):

- Contributors: Tobias Birnbaum and Raees Kizhakkumkara Muhamad (VUB/IMEC), Antonin Gilles (b<>com)
- Changes:
  - Added optional specification of output dir in *nrsh* (may be required for disk-space reasons)
  - Added optional *doDynamic* flag in *nrsh*
  - Disabled "show" in config_files for subj. test holograms
  - *nrsh_video*: Sanitized Fourier/non-Fourier branching code + Fixed minor typo in printout
  - *crop_video*: Added. Similar to join_video but for Breakdancer4k8k which requires spatial cropping of the reconstructions
  - *diff_resize2*: Extended input interface by ratio mode + fixed comments
  - *calcApSizeSimple*: Fixed constraint check for minimal required propagation distance
  - *calcApSizeSimple*: Added error check in case of insufficient diffraction (e.g. BreakDancers4k8k to 2048x2048)
  - *nrsh_video*: Fixed low resolution interface for *wut_disp* etc. + simplification of extranous main-code cases
  - *join_video*: Added
  - *calcApSizeSimple*: Fixed that reconstruction may never exceed target resolution
  - Modified *load_data* and *load_data_auto* to take into account recent updates on *wut_disp_on_axis* data
  - Improved *rgb_align* without quantization

- Added support for *wut_disp_on_axis* in *read_render_cfg, load_data, load_data_auto* and *nrsh_gui*.
- Corrected sign convention for Fourier-Fresnel propagation (*rec_fourier_fresnel*) and aperture size (*aperture_gen_pixel*)
- *nrsh*.m*: Added support for wut_disp_on_axis (Translation + changing dataset to *wut_disp* off axis for remaining reconstruction)
- Added *nrshDR2*: 2D perspective diffraction limited resizing
- Added MatlabEXR .cpp files
- Added *ffmpeg* support for linux/mac-os

Version 4.0 (wg1m89068):

- Contributors: Tobias Birnbaum and Raees Kizhakkumkara Muhamad (VUB/IMEC), Antonin Gilles (b<>com), Manuela Pereira (UBI)
- Changes:
  - Added *ffmpeg* binary
  - Moved *clipping* after *wut_filter* in *nrsh*
  - Added optional parameter *name_prefix* to *nrsh_video, nrsh, nrsh_gui, nrsh_complex* and *nrshDR*
  - Added diffraction limited reconstruction of holograms (*nrshDR* and *diff_resize*)
  - Parameter *save_abs* has been replaced by *save_as_mat* in configuration files
  - Numerical reconstructions can now be saved as amplitude or intensity distributions (*save_intensity* in configuration files)
  - Support for video reconstructions (*nrsh_video*)
  - Support for column vector arguments (*nrsh_video, nrsh, nrsh_gui, nrsh_complex)*
  - Clearer error messages in *nrsh_video, nrsh, nrsh_gui* and *nrsh_complex*
  - Additional (optional) parameter for forward and inverse propagation in *nrsh_complex*
  - Bug fix: added *rec_dist* as a parameter of aperture function
  - Bug fix: saving of reconstructions into MAT files
  - Bug fix: reconstruction of WUT color holograms
  - Bug fix: loading of InterfereIV optical holograms
  - Added third-party library MatlabEXR source code for MEX files building
  - Added new configuration files for b<>com holograms
  - Improved the *rec_asm* function memory usage

Version 3.0 (wg1n88042):

- Contributors: Antonin Gilles (b<>com), Tobias Birnbaum and Raees Kizhakkumkara Muhamad (VUB/IMEC)
- Changes:
  - Support for absolute clipping threshold after reconstruction (*nrsh* and *nrsh_gui*)
  - Filtering of evanescent waves in the Angular Spectrum Method (*rec_asm*)
  - Support for the calculation of complex light waves in the object plane, by-passing non-invertible functions (*nrsh_complex*)
  - Support for inverse Fresnel (*rec_fresnel*) and Fourier-Fresnel (*rec_fourier_fresnel*) transforms, when the reconstruction distance is negative
  - *nrsh_print* has been modified to use configuration files
  - Configuration files are updated to support the reconstruction of high-resolution holograms.
  - Described the light wave propagation to the object plane (Section 1.4 LIGHT WAVE PROPAGATION TO THE OBJECT PLANE) and rendering of high-resolution

holograms for printing applications (Section 1.5 RENDERING OF HIGH-RESOLUTION HOLOGRAMS FOR PRINTING APPLICATIONS)
    - Added a full version history to the user guide
    - Added numerical reconstruction examples for b<>com's Dices200k
    - Minor modifications and fixes to the user guide
    - Used the ISO template for User guide

Version 2.0 (wg1n86036):

- Contributors: Antonin Gilles (b<>com)
- Changes:
    - Support for the reconstruction of high-resolution holograms provided by b<>com and ETRI for printing applications (*nrsh_print*)

Version 1.2 (wg1n85029):

- Contributors: Roberto Corda, Cristian Perra (UNICA)
- Changes:
    - Interfere IV holograms support (experimental)
    - *nrsh* has been renamed in *nrsh_gui*
    - *nrsh_batch* has been renamed in *nrsh*
    - Fixed a bug that did not allow the automatic loading of some holograms from a folder with *nrsh* function

Version 1.1.1 (wg1n84053):

- Contributors: Roberto Corda, Cristian Perra (UNICA)
- Changes:
    - Corrected the messages displayed with *nrsh* regarding the loading of WUT holograms
    - Added error message when in the folder provided to *nrsh_batch* for the automatic hologram loading there are unexpected files
    - Corrected a bug in the synthetic aperture check
    - A new Section (2.1 SYNTHETIC APERTURE) has been added to the user guide
    - Described the messages displayed if incorrect combinations regarding the synthetic aperture are found during the check (Section 2.4 NRSH PROMPT MESSAGES AND AVERAGE TIMES CALCULATION)
    - Minor modifications and fixes to the user guide

Version 1.1 (wg1m84043):

- Contributors: Roberto Corda, Cristian Perra (UNICA)
- Changes:
    - The Neural Network Toolbox is no longer required;
    - The reconstruction process of W.U.T. holograms is now based on the W.U.T FDHrender software, provided by W.U.T. in WG1M83031 [5]
    - The user input and configuration file parameters undergo a checking process before start the reconstructions, in order to reduce the probability of errors during advanced stages of processing; also the synthetic aperture size/position are checked before start the reconstructions
    - The holo_render and holo_render_ng functions are renamed as *nrsh* and *nrsh_batch* respectively;
    - *nrsh_batch* supports the automatic loading of Pleno DB holograms from a folder;

- The user can save the reconstructed image (real-valued) in MAT format;
- The user can set the bit depth of the saved PNG image (8 or 16 bpp only);
- File saving (MAT/PNG) can be disabled;
- The reconstructions can displayed in Matlab windows, during the reconstruction process;
- Configuration files are updated to support the new features.

Version 1.0 (wg1n83060):

- Contributors: Roberto Corda, Cristian Perra (UNICA)
- First version

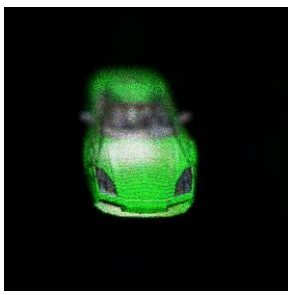
4. RECONSTRUCTION EXAMPLES

In the following, some reconstructions examples are shown. These reconstructions are performed with the configuration files provided in the config_files folder.

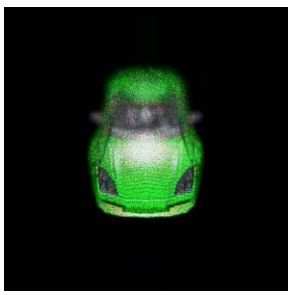
3.1 B-COM: SPECULAR CAR 8K

info = getSettings('cfg_file', 'config_files/bcom/specular_car8k_000.txt', ...
'apertureinpxmode', false, ...
'ap_sizes', 7, ...
'h_pos', -10:10:10, ...
'v_pos', -8:8:8);

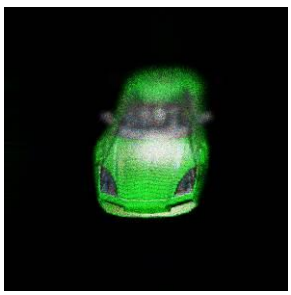
nrsh('', 0.0023, info);



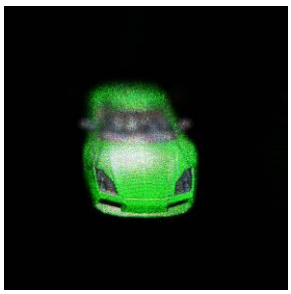
Θ=-10 ϕ=8



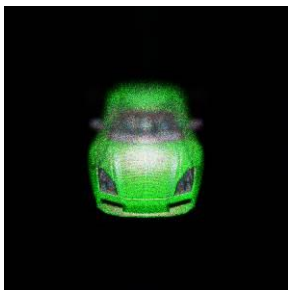
Θ=0 ϕ=8



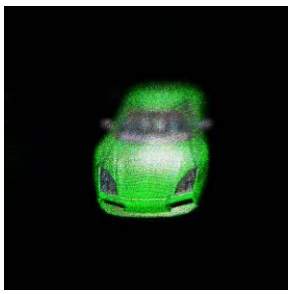
Θ=10 ϕ=8



Θ=-10 ϕ=0

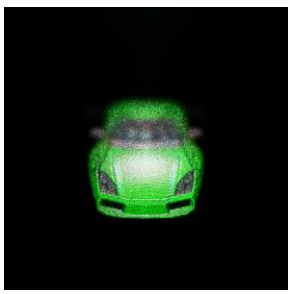


Θ=0 ϕ=0

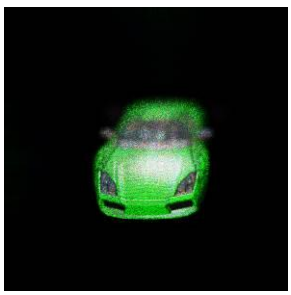


Θ=10 ϕ=0

Θ=-10 ϕ=-8



Θ=0 ϕ=-8

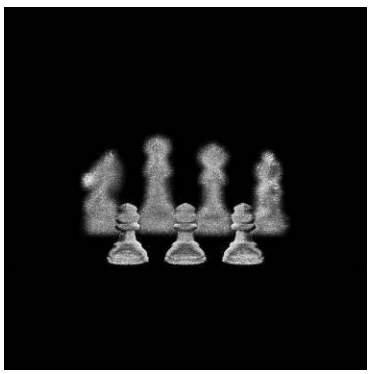


Θ=10 ϕ=-8

3.2 INTERFERE II: CHESS 8KD

info = getSettings('cfg_file', 'config_files/interfereII/chess8kd_000.txt', ...
'apertureinpxmode', false, ...
'ap_sizes', 7);

nrsh('./data/Chess8kd', [0.014:0.001:0.016], info);



rec_dists=0.014

rec_dists=0.015

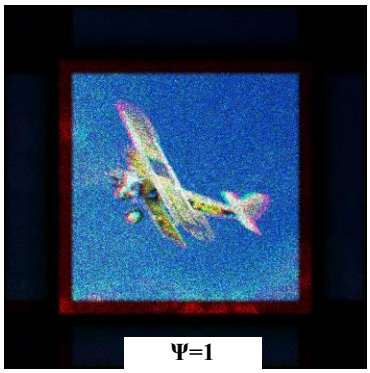
rec_dists=0.016

3.3 INTERFERE III: BIPLANE 16K

load('CGH_Biplane16K.mat');

info = getSettings('cfg_file', 'config_files/interfereIII/biplane16k_000.txt', ...
'dataset', 'interfere', ...
'apertureinpxmode', false, ...
'ap_sizes', [1, 2, 4], ...
'resize_fun', @(x) imresize(x, [2048 1024], 'bilinear'));

nrsh(CGH.Hol, 0.037, info);



Ψ=1

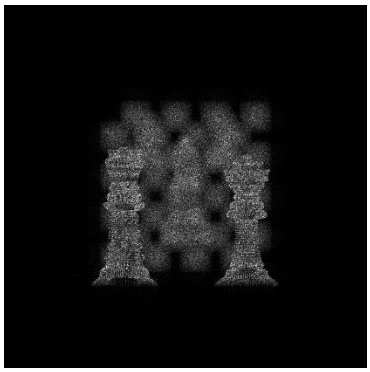
Ψ=2

Ψ=4

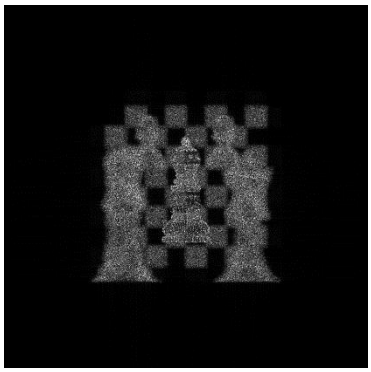
3.4 INTERFERE IV: CHESS2

info = getSettings('cfg_file', 'config_files/interfereIV/chess2_000.txt', ...
'apertureinpxmode', true, ...
'ap_sizes', {[2048, 2048]});

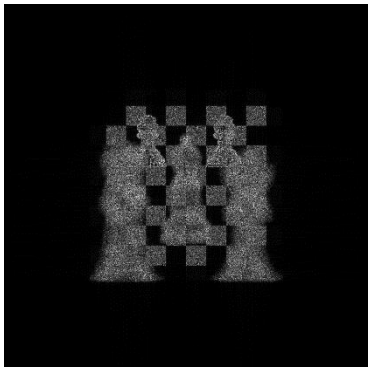
nrsh('./data/Chess2', [0.491, 0.649, 0.806], info);



rec_dists=0.491



rec_dists=0.649

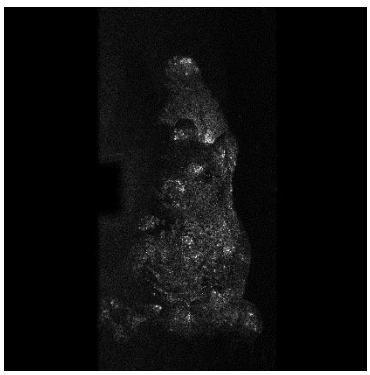


rec_dists=0.806

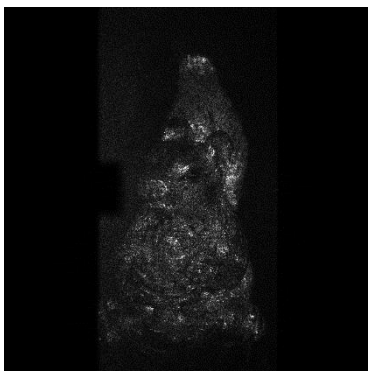
3.5 INTERFERE IV: SQUIRREL

info = getSettings('cfg_file', 'config_files/interfereIV/chess2_000.txt', ...
'apertureinpxmode', true, ...
'ap_sizes', {[1792,1792]}, ...
'h_pos', [-1,0,1]);

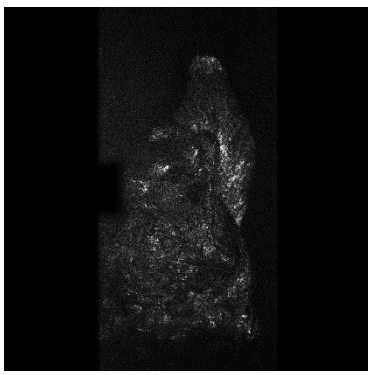
nrsh('./data/Squirrel', 0.535, info);



h_pos=-1



h_pos=0

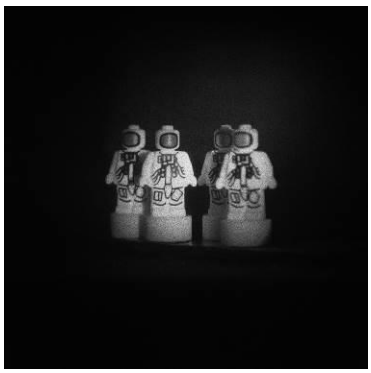


h_pos=1

3.6 EMERGIMG-HOLOGRAIL: ASTRONAUT

info = getSettings('cfg_file', 'config_files/emergimg/astronaut_000.txt');

nrsh ('', -0.1721, info);



rec_dists =-0.1721

3.7 WUT DISPLAY: LOWICZANKA DOLL (PIXEL-BASED MODE)

load('opt_Warsaw_Lowiczanka_Doll.mat');

info = getSettings('cfg_file', 'config_files/wut/lowiczanka_doll_000.txt', ...
'dataset', 'wut_disp_on_axis',...
'ap_sizes', {[2000,3000],[2000, 6000],[0,0]}, ...
'resize_fun', @(x) imresize(x, [2048 1024], 'bilinear'), ...
'h_pos', 1, 'v_pos', 0);

nrsh (dh, 1.06, info);

ap_size=[2000 x 3000]

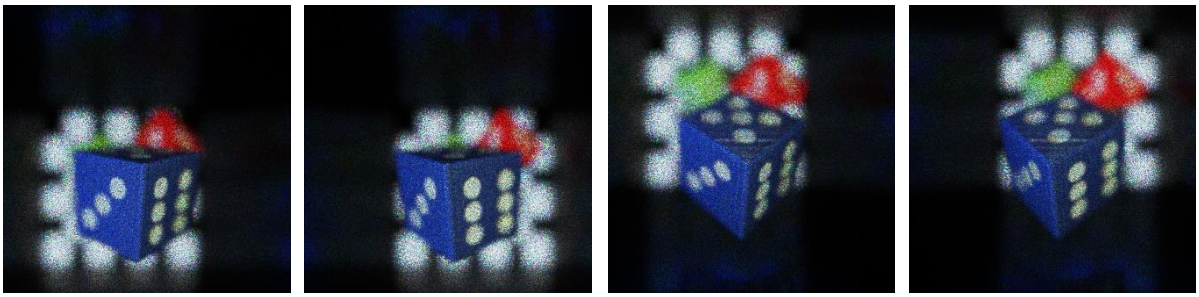
ap_size=[2000 x 6000]

ap_size=[0 x 0]
(full aperture)

3.8 B-COM: DICES16K

info = getSettings('cfg_file', 'config_files/bcom/dices8k_000.txt', ...
'resize_fun', 'DR', ...
'target_res', {[4096,4096]}, ...
'h_pos', [-0.5,0.5], ...
'v_pos', [-0.75, 0.75]);

nrsh('./data/Dices16k', 0.00329, info);



h_pos=-0.5, v_pos=-0.75 **h_pos=0.5, v_pos=-0.75** **h_pos=-0.5, v_pos=0.75** **h_pos=0.5, v_pos=0.75**

3.9 B-COM: DICES 200K (COMPLEX)

info = getSettings('cfg_file', 'config_files/bcom/ dices200k_complex_000.txt', ...
'dataset', ''bcom_print');

nrsh_print('./data/Dices200k', [1 1 4; -1 -1 4; 0 0 4], info);

Viewpoint = [-1, -1, 4] **Viewpoint = [0, 0, 4]** **Viewpoint = [1, 1, 4]**

REFERENCES

[1] Corda, R.; Giusto, D.; Liotta, A.; Song, W.; Perra, C. Recent Advances in the Processing and Rendering Algorithms for Computer-Generated Holography. Electronics 2019, 8, 556.

[2] Gilles, A.; Gioia, P. ISO/IEC JTC 1/SC 29/WG1 M82039, Lisbon, Portugal, January 2019.

[3] Gilles, A.; Gioia, P. ISO/IEC JTC 1/SC 29/WG1 M86034, Sydney, Australia, January 2020.

[4] Oh K.-J., Yeom H.-J., Cheon S., Park J., P. ISO/IEC JTC 1/SC 29/WG1 M86010, Sydney, Australia, January 2020.

[5] Kozacki, T.; Zaperty, W.; Choo, H.; Kim, J. ISO/IEC JTC 1/SC 29/WG1 M83031, Geneva, Switzerland, March 2019.