

# VR we are

## User Manual



Version 4.0, UNRELEASED DRAFT of 1<sup>st</sup> edition  
© 2025 Fortuna Cournot, <https://www.3d-gallery.org>

# Content

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
<b>2</b>	<b>Getting Started .....</b>	<b>5</b>
2.1	<i>System Requirements .....</i>	5
2.2	<i>Installation Step.....</i>	6
2.2.1	Install Tools .....	7
2.2.2	Install ComfyUI.....	7
2.2.2.1	Choose Installation Option .....	7
2.2.2.1.1	Option 1: Windows Portable Package .....	7
2.2.2.1.2	Option 2: ComfyUI Desktop (not regularly tested! Limited support.) .....	7
2.2.3	Install Prerequisites on ComfyUI.....	7
2.2.3.1	Install ComfyUI Custom Nodes .....	7
2.2.3.2	Install ComfyUI Models.....	8
2.2.3.2.1	Upscale Model .....	8
2.2.3.2.2	Depth Model.....	8
2.2.3.2.3	Dubbing Model .....	8
2.2.3.2.4	Controlnet Model <sup>V3.1</sup> .....	8
2.2.4	Install „VR we are“ .....	9
2.2.5	Configure „VR we are“ .....	9
2.2.5.1	Configure for ComfyUI Desktop.....	9
2.3	<i>Setup of the VR headset.....</i>	10
2.4	<i>Skybox VR Player.....</i>	10
2.5	<i>DEO VR Player.....</i>	10
2.6	<i>4XVR Video Player.....</i>	10
2.7	<i>potplayer.....</i>	10
<b>3</b>	<b>System Overview.....</b>	<b>11</b>
3.1	<i>User Interface .....</i>	12
3.1.1	Display Toggles.....	12
3.2	<i>Tool Pipeline.....</i>	12
3.2.1	Customizing Workflow Pipeline .....	13
3.2.1.1	Video Stream Keys.....	14
3.2.1.2	Image Stream Keys .....	14
3.2.1.3	Audio Stream Keys.....	14
3.2.1.4	System Keys .....	14
3.2.1.5	Operators.....	14
3.2.1.6	Values .....	14
3.3	<i>Cleanup .....</i>	14
<b>4</b>	<b>Review and Release System .....</b>	<b>15</b>
4.1	<i>Preparation .....</i>	16
4.2	<i>Edit: Crop, Trim, Snapshots.....</i>	17
4.3	<i>Rating .....</i>	18
4.4	<i>Release.....</i>	19
<b>5</b>	<b>Stages .....</b>	<b>20</b>
5.1	<i>caption.....</i>	20
5.2	<i>scaling.....</i>	20

5.3	<i>fullsbs</i> .....	21
5.4	<i>interpolate</i> .....	21
5.5	<i>singleloop</i> .....	21
5.6	<i>dubbing/sfx</i> .....	21
5.7	<i>watermark/encrypt</i> .....	21
5.8	<i>watermark/decrypt</i> .....	22
5.9	<i>concat</i> .....	22
5.10	<i>check</i> .....	22
<b>6</b>	<b>Slideshow stages</b> .....	<b>23</b>
6.1	<i>slides</i> .....	23
6.2	<i>slideshow</i> .....	23
<b>7</b>	<b>Tasks</b> .....	<b>24</b>
7.1	<i>FFmpeg-based</i> .....	24
7.1.1	vlimit-2160p, vlimit-1080p and vlimit-720p.....	24
7.1.2	hlimit-2160 ,hlimit-4k.....	24
7.1.3	fpslimit-30, fpslimit-15 .....	24
7.1.4	reencode .....	24
7.1.5	pad-16-9, pad-16-9-sbs .....	24
7.1.6	credit-vr-we-are, credit-no-sbs .....	24
7.1.7	desaturate, incsaturation .....	24
7.1.8	split-1m .....	25
7.1.9	createsilence, stripaudio .....	25
7.1.10	swapsides .....	25
7.2	<i>ComfyUI-workflow-based</i> .....	25
7.2.1	recolor .....	25
<b>8</b>	<b>Customization</b> .....	<b>26</b>
<b>9</b>	<b>Stereoscopic - A Full Stereo Side By Side Converter</b> .....	<b>27</b>
9.1	<i>Background</i> .....	27
9.2	<i>Depth Map</i> .....	27
9.3	<i>SBS Conversion Algorithm</i> .....	28
<b>10</b>	<b>Advanced Features</b> .....	<b>30</b>
10.1	<i>Topaz Video AI</i> .....	30
<b>11</b>	<b>Troubleshooting</b> .....	<b>31</b>
<b>12</b>	<b>Acknowledgements</b> .....	<b>32</b>
	<b>Appendix: Default Workflow Pipeline.....</b>	<b>33</b>



## 1 Introduction

Welcome to „VR we are“, an immersive productivity and creativity software designed to bring the familiar multimedia content of videos and images into the world of a Virtual Reality (VR) environment, using Artificial Intelligence (AI) models on the local computer, and most of the processing can be done offline.

Key feature is conversion of 2D images and videos into full stereo side-by-side (SBS) left-right (LR), which can be visualized with viewers in VR headsets or glasses on 3D-capable TV displays.

Some other features can be used even without targeting a VR device, such as capture meta data, upscaling, frame interpolation, various ffmpeg tasks, dubbing or creating slideshow videos from images.

A central principle of the application is to use conventions over configuration. That said I try to keep *required* configuration to a minimum by providing default settings wherever possible.

## 2 Getting Started

This chapter explains the basic system requirements, installation steps, and setup of the VR headset.

### 2.1 System Requirements

The AI tools require a modern computer to perform well. Nevertheless, small media files can be handled on older hardware.

- Operating System: Windows 10 or higher.
- RAM: 16 GB minimum, 64 GB or higher recommended.
- VRAM: Not required. 16 GB or higher recommended.
- GPU: Not required. CUDA-capable GPU and setup recommended.
- File system: 40 GB free space minimum, 200 GB or more recommended. More than 10 GB free space is required to run.

Software Versions:

- ComfyUI 0.3.59
- FFmpeg git-2020-08-31-4a11a6f
- Exiftool 13.33
- Topaz VideoAI 6.0.4

Supported media:

- Although the tool accept any media that Ffmpeg, OpenCV or ComfyUI can handle, I mainly support
  - mp4 with h264 codec
  - webm
- Other file types may work depending on the stage, but are not consistently supported:
  - jpg
  - png
  - webp
- Media height should be even (divisible by 2).



## 2.2 Installation Step

The installation goes in the following order:

1. Install Tools
2. Install ComfyUI
3. Install Prerequisites on ComfyUI
4. Install „VR we are“
5. Configure „VR we are“

## 2.2.1 Install Tools

If you already have them, I recommend updating, since old version may cause problems.

- Exiftool. Rename exe (strip -k) for usage in CLI and add to PATH environment variable, it will be detected later, and simplifies setup.  
<https://exiftool.org/>
- FFmpeg. Add to PATH environment variable, it simplifies setup.  
<https://ffmpeg.org/>
- Git Bash (for Windows).  
<https://gitforwindows.org/>
- Topaz Video AI (TVAI) – You need current version (6); future versions will stop supporting CLI. You must configure paths later.

Change setup after installation, by editing the following variables in config.ini:

EXIFTOOLBINARY, FFMPEGPATHPREFIX, TVAI\_BIN\_DIR, TVAI\_MODEL\_DATA\_DIR and TVAI\_MODEL\_DIR.

## 2.2.2 Install ComfyUI

### 2.2.2.1 Choose Installation Option

#### 2.2.2.1.1 Option 1: Windows Portable Package

Download a 7z file from release 0.3.58 to 0.3.59 on GitHub:

<https://github.com/comfyanonymous/ComfyUI>

It includes Python. Execute run\_cpu.bat or the nvidia ones if you have that GPU.

#### 2.2.2.1.2 Option 2: ComfyUI Desktop (*not regularly tested! Limited support.*)

You need to install Python 3.12+ and ComfyUI Desktop:

<https://www.python.org/downloads/>

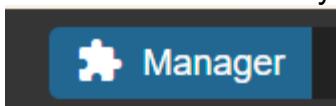
<https://www.comfy.org/>

## 2.2.3 Install Prerequisites on ComfyUI

### 2.2.3.1 Install ComfyUI Custom Nodes

In ComfyUI, if not already there, install the Manager. (Download "Code" ZIP and move folder into ComfyUI\_windows\_portable\ComfyUI\custom\_nodes.

Then restart the ComfyUI server and you will see this icon:



Now wait for the Manager to update the ComfyRegistry (watch logs), so the custom node database is up to date.

Custom Nodes Manager

Now, you can use the ComfyUI Manager to install other custom node packs. It will download most of them from GitHub:

- comfyui\_controlnet\_aux 1.1.0
- ComfyUI-Custom-Scripts 1.2.5
- comfy-mtb 0.5.4
- ComfyUI-Crystools 1.27.3
- ComfyUI-Florence2 1.0.6
- ComfyUI-VideoHelperSuite 1.7.4
- ComfyUI-Frame-Interpolation 1.0.7

Then install ComfyUI-MMAudio manually:

Download from <https://github.com/kijai/ComfyUI-MMAudio>, then install to ComfyUI/custom\_nodes folder, then execute in the bash shell:

```
./python_embeded/python.exe -m pip install accelerate
./python_embeded/python.exe -m pip install -I wcwidth
./python_embeded/python.exe -m pip install --upgrade numpy==2.2
```

```
# Locate and switch to installation folder:
cd ComfyUI_windows_portable
# Execute install script
./python_embeded/python.exe -m pip install -r ComfyUI/custom_nodes/ComfyUI-MMAudio/requirements.txt
# create model directory
mkdir -p ComfyUI/models/mmaudio
```

### 2.2.3.2 Install ComfyUI Models

#### 2.2.3.2.1 Upscale Model

Download 4x-realesrgan-x4plus manually from <https://openmodeldb.info/models/4x-realesrgan-x4plus> and place it at ComfyUI\models\upscale\_models\RealESRGAN\_x4plus.pth  
Or use and configure other upscale models in config.ini later.

#### 2.2.3.2.2 Depth Model

With version 3.0 depth\_anything\_v2\_vitb.pth is preconfigured, since the giant model is a performance killer. If you want to use it anyway then download Depth-Anything-V2-Giant manually from <https://huggingface.co/likeabruh/depth Anything v2 vitg/tree/main> and place it at ComfyUI\custom\_nodes\comfyui\_controlnet\_aux\ckpts\depth-anything\Depth-Anything-V2-Giant\depthAnything\_v2\_vitg.pth and configure it later in config.ini

#### 2.2.3.2.3 Dubbing Model

Please follow this Guide on Github to install MMAudio models:

<https://github.com/kijai/ComfyUI-MMAudio?tab=readme-ov-file#installation>

In short:

Download the fp16 variants of the models from huggingface (4 files):

[https://huggingface.co/Kijai/MMAudio\\_safetensors/tree/main](https://huggingface.co/Kijai/MMAudio_safetensors/tree/main)

and move them to ComfyUI\models\mmaudio

#### 2.2.3.2.4 Controlnet Model<sup>v3.1</sup>

For the recolor task the following model (from ComfyUI model manager) is required:  
stabilityai/control-lora-recolor-rank256.safetensors

## 2.2.4 Install „VR we are“

Use ComfyUI Manager to install the latest version of „VR we are“ (= `comfyui_stereoscopic`). When restarting the ComfyUI server, watch the log files for any errors<sup>1</sup>.

From the ComfyUI/custom\_nodes/comfyui\_stereoscopic folder you can create desktop shortcuts from the two batch files, by dragging them with the right(!) mouse key on the desktop. They will call shell scripts. Instead you can still call the shell scripts by hand from git bash; this can be necessary to investigate errors. The desktop shortcuts can be configured with icon and you can choose to minimize the window shell.

Now call `demon.bat` or `daemon.sh` and the installation will complete and be tested (takes up to 3 minutes). You will notice a small guy window to popup as well. If the tests went well, you can do the configuration. Close the daemon by pressing **CTRL + C** in the bash shell.

To call the script manually, open a new Git Bash Shell, and drag the script there or enter:

```
# Locate and switch to installation folder:  
cd ComfyUI_windows_portable/ComfyUI  
# Execute background task  
./custom_nodes/comfyui_stereoscopic/daemon.sh
```

## 2.2.5 Configure „VR we are“

The properties are located at `ComfyUI\user\default\comfyui_stereoscopic\config.ini`

I recommend to complete and check and complete the values for the following keys:  
`EXIFTOOLBINARY`, `DESCRIPTION_LOCALE`, `ISO_639_2_CODE`, `WATERMARK_LABEL`

If TVAI is installed provide `TVAI_BIN_DIR`, `TVAI_MODEL_DATA_DIR`, `TVAI_MODEL_DIR` (request CLI instructions Video AI - Nvidia).

If you don't want to the pipeline, e.g. it is confusing you; disable auto forwarding by setting `PIPELINE_AUTOFORWARD` to 0. Auto forwarding is a powerful feature; you can edit the pipeline in `ComfyUI\user\default\comfyui_stereoscopic\rebuild_autoforward.sh` which is executed every time the daemon is started.

Many more properties can be modified there. If the tool fails, delete the config file and the generator will generate a new one with default values.

You need to restart `daemon.sh` after changing the configuration. You do not need to restart the ComfyUI server.

### 2.2.5.1 Configure for ComfyUI Desktop

For ComfyUI Desktop, the port needs to be changed, since the scripts are using the default port of ComfyUI Portable. Go either to Settings->Server Config, and set the value for the port to 8188, or change the value of the property `COMFYUIPORT` in `config.ini` to the ComfyUI Desktop port (8000, look into Settings in ComfyUI Window). Sometimes when you close the window, the new instance will allocate a different port, like 8002, because the old process has not terminated.

---

<sup>1</sup> The latest ComfyUI requires `opencv-python>=4.10.0.84` and raise an import error for `cv2` package, in case you try to use it with version 3.0 of `comfyui_stereoscopic` module, which can be changed in `requirements.txt` file.

## **2.3 Setup of the VR headset**

### **2.4 Skybox VR Player**

Commercial application. Supports many platforms. We have tested it on Meta Quest 3.

To simply the configuration follow their guide: How to Adjust 2D/3D/VR Video Formats, e.g. by putting all videos and images under a folder with the name "fullsbs".

*Known problems:* To view images, the curved mode must be deactivated (bugged). Projection problems with height adjustment at high zoom.

### **2.5 DEO VR Player**

Download for free. Supports many platforms. We have tested it on Meta Quest 3, HTC vive pro 2.

To simply the configuration follow their guide: Naming convention, e.g. by adding "\_SBS\_LR" to the end of the file name.

*Known Problems:* Projection problems with height adjustment at high zoom. No controller mapping for pitch. Subtitle track not selectable.

### **2.6 4XVR Video Player**

Another Commercial application.

*Known problems:* Auto-detection of aspect ratio currently does not work (support pending). The player does not support images. Manual setup per file required to 3D FSBS if aspect unusual. No height adjustment. Subtitles not supported.

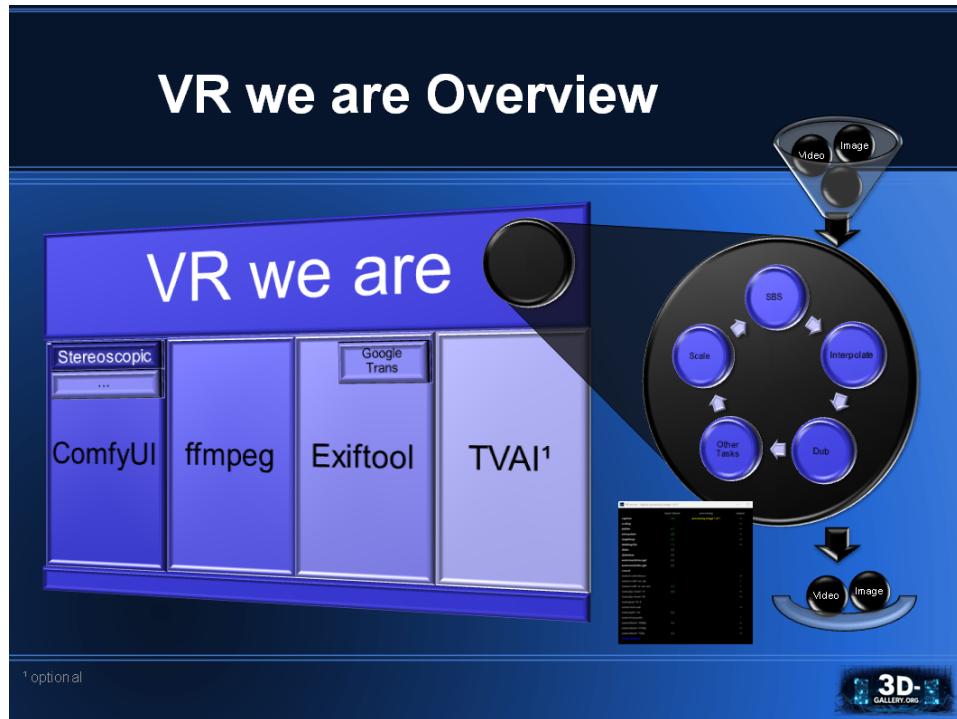
### **2.7 potplayer**

Download for free. Requires an active connection link to the computer. Can be used as video source for other VR media players that run on the Quest 3, such as Bigscreen.

*Known problems:* The player does not support images. Other video aspects than 16:9 are not suitable for SBS display, but you can use the task *pad-16-9-sbs* to counteract this problem.

### 3 System Overview

The following picture illustrates the building blocks of „VR we are“:



„VR we are“ is using other software as foundation:

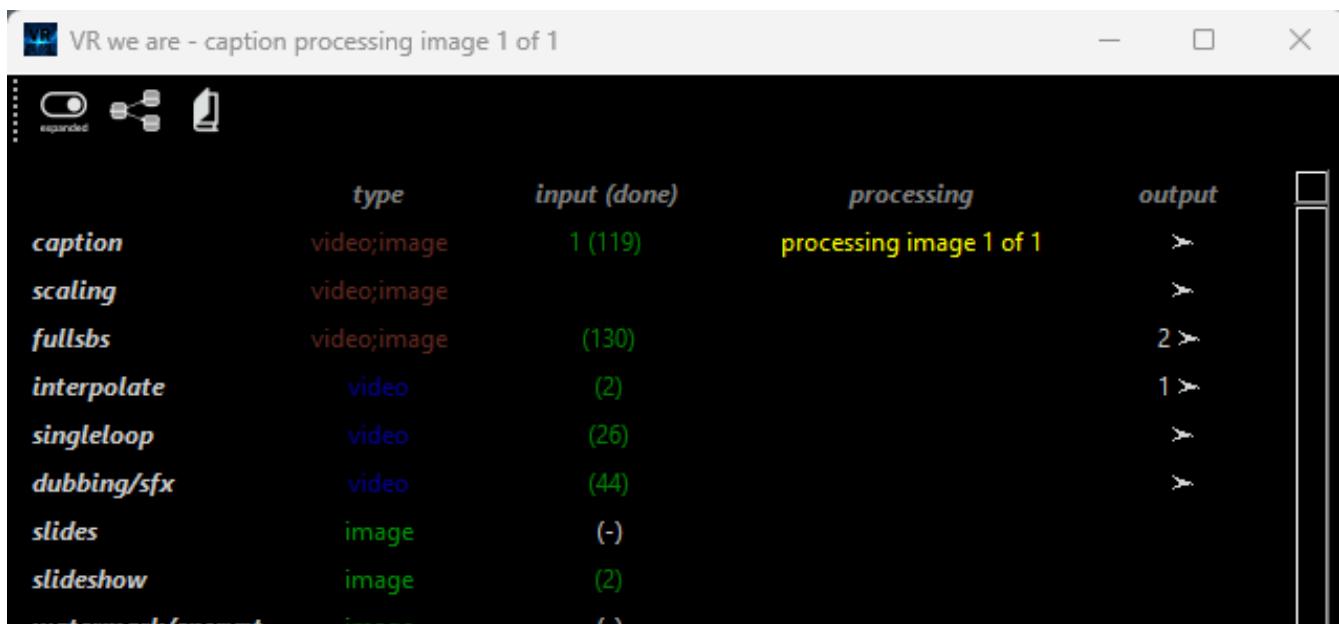
- ComfyUI is building foundational open source software for the visual AI space.  
„VR we are“ uses it as distribution and execution platform
  - Stereoscopic is a custom node package for ComfyUI containing the „VR we are“ software. For the custom node I got help from iablunoshka, responsible for high performance of the SBS converter. Our first tests had been made with the nodes of SamSeen.
- FFmpeg is a command line tool providing a multimedia framework for video and image manipulation.
- Exiftool is a command line tool for reading and editing multimedia meta information.
  - Google Trans is an optional service, requiring to be online, to translate text (into the own locale).
- Topaz Video AI (TVAI) is an optional professional product, when available used. It offers a massive speed and quality boost on scaling and video interpolation (frame rate increase).
- Git Bash package, an application for Microsoft Windows environments, which provides an emulation layer for a Git command line experience, required to execute „VR we are“.

„VR we are“ waits for multimedia files to be placed in input funnels (file folders) for processing. Per default the files are processed in a non-linear pipeline, landing in output baskets (file folders). Pipelining through the stages can be customized or even completely turned off.

**Important:** Never place your original files there; only copies!

### 3.1 User Interface

A small graphical user interface is started with the daemon. One duty is to show the working status with a table of the stages and the file count of folders:



	type	input (done)	processing	output
<i>caption</i>	video;image	1 (119)	processing image 1 of 1	➤
<i>scaling</i>	video;image			➤
<i>fullsbs</i>	video;image	(130)		2 ➤
<i>interpolate</i>	video	(2)		1 ➤
<i>singleloop</i>	video	(26)		➤
<i>dubbing/sfx</i>	video	(44)		➤
<i>slides</i>	image	(-)		
<i>slideshow</i>	image	(2)		
<i>watermark/concat</i>	image	(-)		

The arrow at output of stages indicates that forward rules exists that may affect the files; when the color turns green, it indicates that forward rules have been applied and the files will stay.

#### 3.1.1 Display Toggles

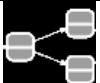
 collapsed	Per default stages are collapsed. Only stages with relevant file counts or while active are displayed.
 expanded	In Expanded state all stages are displayed, and supported stage input types are shown: Video, Image or both

### 3.2 Tool Pipeline

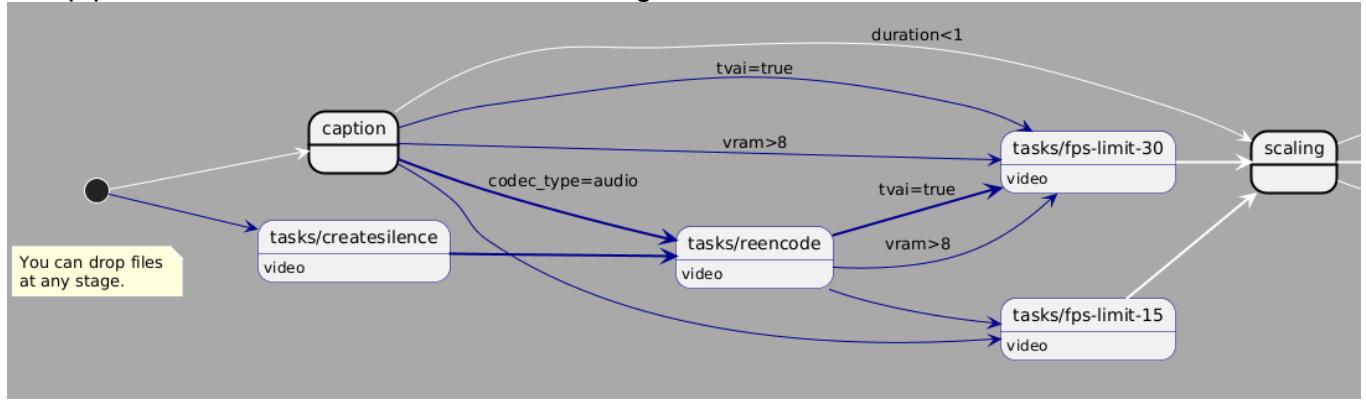
To simplify things, the tool offers a video and image processing pipeline with auto-forward from stage output to another stage input. With release 3.0 it is no longer linear. It can be customized with care.

You just need to place files in input folder of a stage, and pick the results up from the final output folders. No more need for common users to create own workflows or other CLI command and shell scripts.

If you get confused by auto-forwarding, just disable it by setting in config.ini  
PIPELINE\_AUFTFORWARD=0

	The current pipeline can be viewed in a dialog by hitting this button.
---	--

The pipeline is visualized as a UML state diagram:



See appendix for a full display of the default workflow pipeline.

### 3.2.1 Customizing Workflow Pipeline

Auto-forwarding is by default active. You can deactivate it by setting PIPELINE\_AUTOFORWARD=0 in ComfyUI\user\default\comfyui\_stereoscopic\config.ini

The workflow is initialized every time the VR-we-are daemon is started, by reading the configuration defined in a YAML file at ComfyUI\user\default\comfyui\_stereoscopic\autoforward.yaml. The file is created with defaults, if it does not exist. It contains the forwarding rules and conditions you can see in the state diagram above.

On daemon start, from this configuration *forward.txt* files in the output stages are created. Media files that land there will be checked against these rules for transfer from output to input stage. The rules are executed in order and the first rule that matches is executed. Editing the *forward.txt* files is possible, in case you just need a temporary change while the daemon is running.

Every rule may have transition conditions. These are rule details, which must all match for the rule to be executed. Conditions are optional. There is an invisible rule that is always checked: rules are only executed if the media type (video or image) is supported by input and output stage.

You can edit the yaml file to permanently modify the pipeline. Syntax is:

```
---  
forwards:  
  - from: STAGEPATH  
    rules:  
      - rule:  
          to: STAGEPATH  
          conditions:  
            - CONDITION
```

YAML list items start with a dash and can be repeated.

Conditions are optional

Where STAGEPATH is a valid stage path like scaling, tasks/fps-limit-15 or tasks/\_myjob

A CONDITION has the syntax KEY OPERATOR VALUE , e.g. width < 3841

### **3.2.1.1 Video Stream Keys**

They come from Ffmpeg / ffprobe:

bit\_rate, width, height, r\_frame\_rate, duration, nb\_frames, display\_aspect\_ratio(multiplied by 1000)

### **3.2.1.2 Image Stream Keys**

They come from Ffmpeg / ffprobe:

width, height

### **3.2.1.3 Audio Stream Keys**

They come from Ffmpeg / ffprobe. This key is special and required to detect audio stream exists (see default config).

codec\_type

### **3.2.1.4 System Keys**

vram(GB), tvai(true/false), sbs(true/false)

### **3.2.1.5 Operators**

= != > <

### **3.2.1.6 Values**

Values must always be integer or text. floats are not supported. Numbers or expressions from ffprobe are truncated (display\_aspect\_ratio is multiplied before).

## **3.3 *Cleanup***

Processed files in input folders are copied into a done subfolder causing a lot of file copies. These folders are cleared when the daemon is restarted. At installation time (once) *.nocleanup* files are created in this folder. If they exist, the files are excluded from cleanup. You can create and delete this *.nocleanup* files as you need. This system is one of the reasons you should never place your original files in input folders. Another reason is, that the application is renaming files. Try to avoid using filenames with non alpha-numeric characters [A-Za-z0-9], especially if they come from untrustworthy sources. I do some checks but can't guarantee it is bullet proof.

## 4 Review and Release System

The User Interface offers a review system, to integrate the txt2img and img2vid techniques into the workflow pipeline. Images and videos from AI are usually created in batches, and must be evaluated and filtered in the meantime, before continuing in your workflow. This can be a time consuming task, which is supported by „VR we are“. The user interface offers a three step process, starting with files supplied in the incoming folder (see chapter 5.10):



substage check/rate:

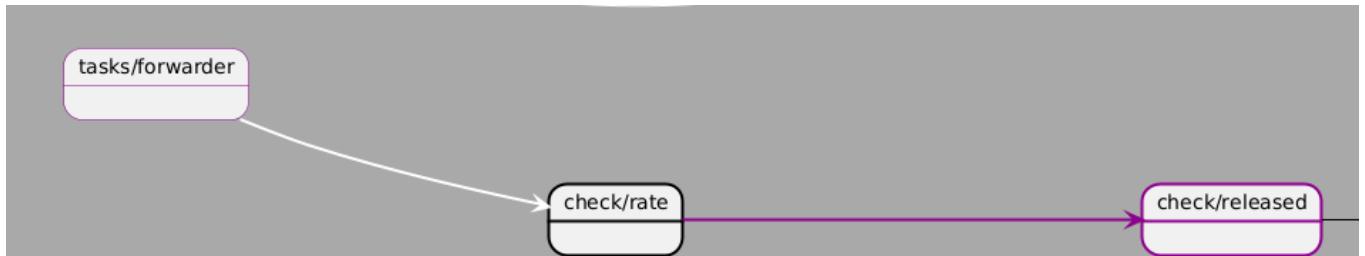
1. Edit: Create copies of cropped images or videos. Create copies of trimmed videos. Create snapshots from videos. Delete scrap.
2. Rate: Categorize images or videos with a 5-star rating system as used by the Windows OS (includes tagging) and forward to next substage.

substage check/released:

3. Delete or release file sets per rating class and forwarding them to output folder (see chapter 5.10).

## 4.1 Preparation

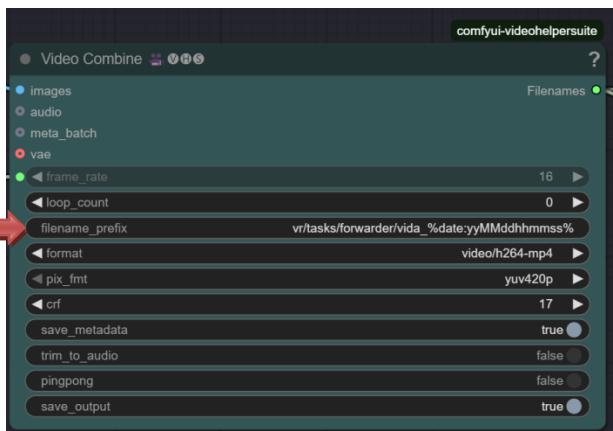
This system is designed to work with your txt2img and img2vid workflows. Save the output of your ComfyUI workflows directly to the *output/vr/tasks/forwarder* folder, if you need no backup. Or if you prefer external tools, copy your media there.



*tasks/forwarder* is a special stage prepared to forward any file to *input/vr/check/rate*.<sup>2</sup> Of course you can save files directly in workflows to the input folder, but you need to use special nodes that allow this.

Ensure your files have unique names, to avoid overwriting old files and lose them!

The following popular custom node allows generating timestamps in the filename. Here an example generates different name every second, what is any case I know sufficient.



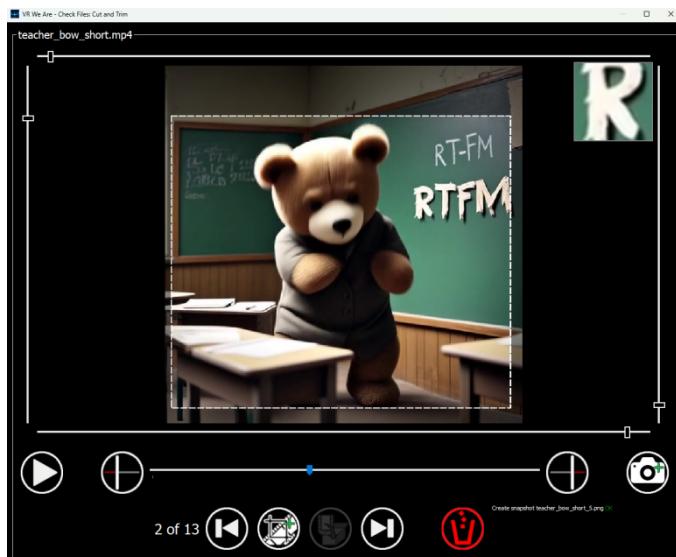
E.g.:

`vr/tasks/forwarder/vida_%date:yyMMddhhmmss%`

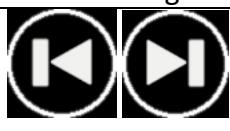
<sup>2</sup> Notice: The forwarder may already try while files are not completely written, causing multiple rename log entries.

## 4.2 Edit: Crop, Trim, Snapshots

Crop & Trim focuses on a quick review of the files, delete crap or create edited copies.



The following controls allow to navigate through the files:



Go to previous or next file in input/vr/check/rate

Some actions can be done with each file

	Creates a copy of the video with the selected crop and trim. <sup>3</sup>
	Moves the file away to <i>input/vr/check/rate/done</i> for later cleanup. No a substitute for rating - this action can only be used after a copy or snapshot of the current media has created, so this raw file is no longer be subject to be rated. This brings you to next file that will be usually the copy.
	Deletes the current file. Allows you to get rid of crap quickly. Shows next file. If you made copies before you will see them, now.

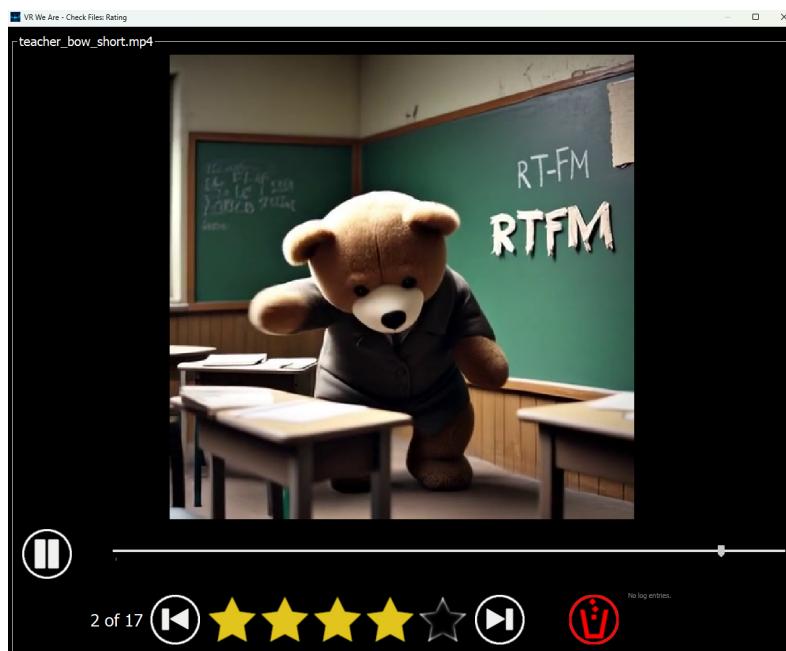
For videos one more set of controls appear:

	Videos are continuously played in loop. This control allows pausing and continuing playing. Some operations require pausing the video.
	Here you see the video playing. While paused, you can select the frame. If trimmed, the loop is adjusted to the trim.
	Selecting video trim at the selected frame to left or right. The trim can be applied by the  action and previewed by starting the playback.
	Creates a snapshot from video at the selected frame position.

<sup>3</sup> Video output height is always (FFmpeg issues with odd numbers).

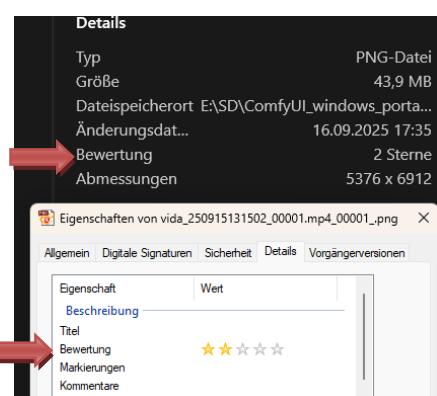
### 4.3 Rating

Rating is separated from Edit. This is your judgement about the quality of media, as preparation for the next stage, where these categories can be forwarded, archived or deleted in a few steps. It does not check for any existing rating.



	Rate the files between 1 and 5. They will be moved to subfolder 1 to 5, below output/vr/check/rate. If Exiftool is present the rating is applied to meta data and visible in OS Windows.
	Deletes this file.

Rated files get metadata attached if Exiftool is properly configured in config.ini. This has been tested with mp4 and png under Windows 11 and may not work with other formats.



## 4.4 Release

Releasing allows to forward, archive or delete a bunch of files by classification category, previously done with rating. It inspects subfolder 1 to 5 at output/vr/check/rate. Click a buttons below a category:



File Categories are displayed by star icons representing on the status of each category:

	No files in this category. Nothing to do.
	Decision required, see actions below.
	Files have been archived. Action completed.
	Files have been deleted. Action completed.
	Files have been forwarded. Action completed.

Possible actions on the lowest and highest outstanding decision to be made:

	Archive all files in this category by moving them to folder <code>input/vr/check/released/done</code>
	Deletes all files in this category.
	Forward files in this folder to <code>output/vr/check/released</code> . If there is <code>forward.txt</code> file, it will be further forwarded to the stages of your choice. Hint: Use first a rule with the condition <code>[duration&lt;1]</code> to forward images, then a rules to forward videos to a different stage. See chapter 3.2.1 for information about how to customize the pipeline.

Hint: You may want to forward and continue processing your best files and come back later, maybe with more rated files, keeping this like a backlog.

## 5 Stages

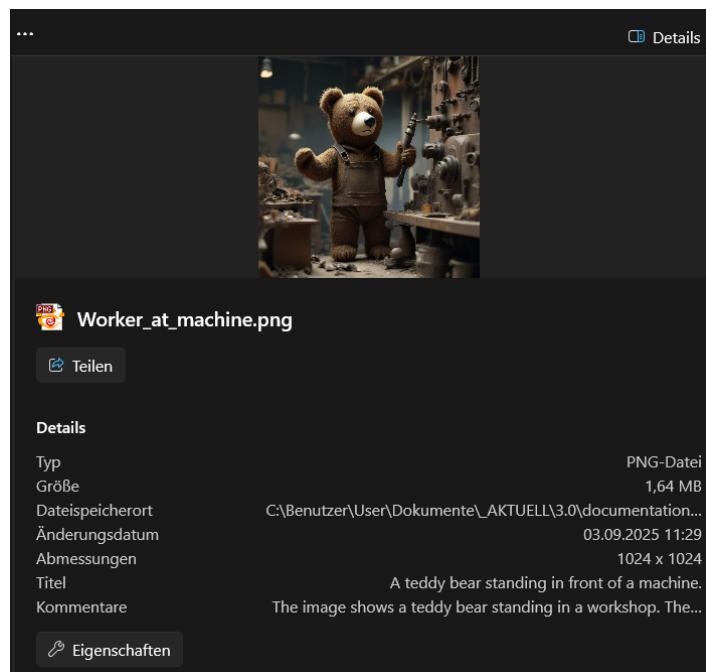
The following stages are the basic operation „VR we are“ can perform:

### 5.1 caption

Adding title, description to videos and images using analysis of first frame.<sup>4</sup> Workflow and prompt are removed.

Translates descriptions if online. They are displayed and editable in the Windows Explorer in Windows 11 (only displayed in Windows 10). Forwards meta data of videos and images through stages , strips off prompts and workflows. To enable, change language or turn off translation please edit the config.ini parameters EXIFTOOLBINARY, DESCRIPTION\_LOCALE, ISO\_639\_2\_CODE.

Under Windows 11, title and comments can been seen in the preview ...



... and in the property details:



### 5.2 scaling

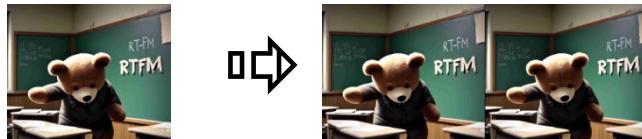
Upscales the video and image with an AI-Model by a factor of 4 or 2 depending on the current size, some rules and the configuration, using a combination of upscale model and mixing with the original input. High resolution input is required by the SBS converter to produce better

<sup>4</sup> The Windows OS requires to remove some Keys and their Metadata.

results at the “edges” of the depth images. Restricted use on old hardware by pipeline conditions. For low quality input and long videos I recommend using tools first. For videos it operates on frames, why fps-rate is limited in the pipeline first.  
Important config.ini parameters: UPSCALEMODELx4, UPSCALEMODELx2.  
TVAI requires model configuration.

### 5.3 *fullsbs*

Convert video and image to side-by-side left right, full width to be used in a viewer on your VR display. Stereoscopic is a main feature of VR we are; please read more about it in chapter 9.



For best results input should have resolution of at least 1MP or higher. For videos it operates on frames, why fps-rate is limited in the pipeline first.

Important config.ini parameters: SBS\_DEPTH\_SCALE, SBS\_DEPTH\_OFFSET, SBS\_DEPTH\_BLUR\_RADIUS\_VIDEO, SBS\_DEPTH\_BLUR\_RADIUS\_IMAGE, DEPTH\_MODEL\_CKPT

### 5.4 *interpolate*

Doubles the video frame rate using AI model. It is a very time-consuming operation. Pipe-line default is self-looping up to 60fps.

config.ini parameters: TVAI requires model configuration.

### 5.5 *singleloop*

Appends a reverse version to a video, so it can be looped. Useful for short videos that have no forward direction in time. I recommend using this before dubbing videos.

### 5.6 *dubbing/sfx*

Designed to add sound (effects) to videos using AI model. Additional prompts can be added. Be aware, that this might be out-of-sync and contain strange parts. It is a tool for you to play with and quickly get some sound. Prompts are injected from ComfyUI\user\default\comfyui\_stereoscopic\dubbing\_sfx\_positive.txt and dubbing\_sfx\_negative.txt.

Seeds are random. Repetitive calls (if auto-forward is deactivated) may produce new sound results, with ascending numbers, for the user to choose.

Long videos >20 seconds, especially slideshows, generate sound changes every 5 seconds, so every slide has its own sound.

### 5.7 *watermark/encrypt*

encode forensic watermark on images, and keep the original image in store under ComfyUI\user\default\comfyui\_stereoscopic\watermark.

Important config.ini parameters:

WATERMARK\_LABEL (maximum 17 characters) containing your name

WATERMARK\_SECRETKEY: Your personal integer used to encrypt the image. Without this key nobody can decrypt the watermark. The key is stored as folder name in the store, so it can't get lost and the key can be changed on demand, e.g. after some period of time.

File watermark\_background.png next to config, contains the watermark image where your name is printed at top:



You replace the image, but the filename is fixed.

### 5.8 *watermark/decrypt*

decode forensic watermarking from image against stored version and return the watermark image.

Returns a ghost image of the watermark that had been invisibly encoded in your images before:



### 5.9 *concat*

Can be used in combinations with tasks/split-1m or singleloop to put parts or loops, which share the same first (and last) frame, to a complete or longer video. For loops I recommend using this before dubbing.

### 5.10 *check*

The review operation is backed up by special stages under the stage name *check*, which can be integrated into the forward pipeline. Input is at *check/rate* and output at *check/released*.

## 6 Slideshow stages

Slideshow stages are outside of the normal pipeline. Turn auto-forwarding off by temporarily setting PIPELINE\_AUTOFORWARD to 0 in config.ini.

Use stages in the following order:

- scaling (optional)
- slides
- slideshow
- fullsbs, from here you may start auto-forwarding again
- interpolate (optional)
- dubbing/sfx (optional)

Finally set PIPELINE\_AUTOFORWARD to 1.

### 6.1 *slides*

Want to see your images, maybe from vacation, in VR, but are annoyed about the way your VR viewer is displaying them? Use this to create 4K slide images (scale + padding) as preparation to make a video. Take the images from the p. Then move all slides to the slideshow.

### 6.2 *slideshow*

Put slides from slides stage into a slide show, 5s each image including an 1s SBS-compatible transition.

## 7 Tasks

Tasks are standardized stages based on a blueprint. The user can duplicate and customize them. He can also replace them by modifying the pipeline.

### 7.1 FFmpeg-based

#### 7.1.1 vlimit-2160p, vlimit-1080p and vlimit-720p

Ensures the height of a media is limited for performance reasons.

#### 7.1.2 hlimit-2160 ,hlimit-4k

Ensures the width of a media is limited for publication reasons.

#### 7.1.3 fpslimit-30, fpslimit-15

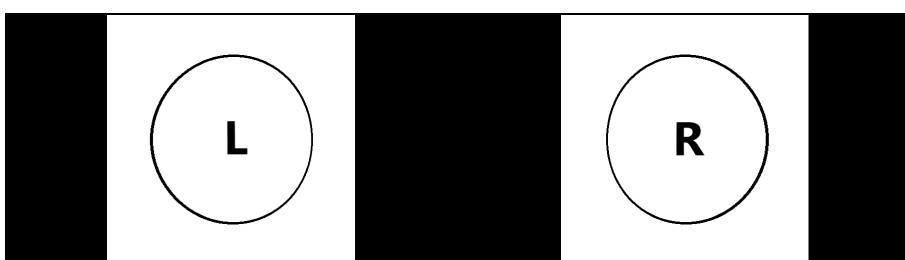
Ensures the frame rate of a video is limited for performance reasons.

#### 7.1.4 reencode

Ensures the media is well defined for processing.

#### 7.1.5 pad-16-9, pad-16-9-sbs

Pads the width of the media with black color so it gets aspect 16:9. For SBS aspect is per side. Result:



This can be handy to display it with viewers that have problems to handle full SBS display of other aspects.

#### 7.1.6 credit-vr-we-are, credit-no-sbs

Adds credits to a video. The credit video must be loopable. This can be used a template to replace it with your own signature.



#### 7.1.7 desaturate, incsaturation

Manipulating saturation.



### **7.1.8 split-1m**

Use this to split long videos into segments of about 1 minute length as first step. After jobs for all segments are done, use the concat stage to merge them together.

### **7.1.9 createsilence, stripaudio**

Adds silent audio to video or strips audio off from video.

### **7.1.10 swapsides**

Swaps left right in SBS video or image. Useful for people who want to create content for cross-eyed stereo.



## **7.2 ComfyUI-workflow-based**

### **7.2.1 recolor**

Coloring an grayscale image using AI.



## 8 Customization

Users can create their own tasks. For now, the blueprint ffmpeg-v2v can be used. You need to have sufficient knowledge of FFmpeg to do.

Just copy a template from ComfyUI\custom\_nodes\comfyui\_stereoscopic\config\tasks to ComfyUI\user\default\comfyui\_stereoscopic\tasks

E.g. hlimit-4k.json and give it an appropriate name.

Example: We want to scale videos down by factor 2. We name it halfsize.json.

To produce what we want we change filter in the json file, so the file looks like that:

```
{  
    "version": "1",  
    "blueprint": "ffmpeg-v2v",  
    "options": "-filter:v scale=iw/2:-2",  
    "input": "video",  
    "output": "video",  
}
```

Now restart the daemon and a new stage appears. User stages have an underscore appended to the task name, so the input folder is ComfyUI\input\vr\tasks\\_halfsize

It will also appear in the status view of the user interface:



It is not integrated in pipeline, unless you do so. Just drop files in the input folder and see how they are processed.

## 9 Stereoscopic - A Full Stereo Side By Side Converter

### 9.1 Background<sup>5</sup>

Stereoscopy, also called stereoscopics or stereo imaging, is a technique for creating or enhancing the illusion of depth in an image by means of stereopsis for binocular vision. Stereoscopy creates the impression of three-dimensional depth from a pair of two-dimensional images. The left image is presented to the left eye and the right image is presented to the right eye. When viewed, the human brain perceives the images as a single 3D view, giving the viewer the perception of 3D depth. However, the 3D effect lacks proper focal depth, which gives rise to the vergence-accommodation conflict. Human vision, including the perception of depth, is a complex process, which only begins with the acquisition of visual information taken in through the eyes; much processing ensues within the brain, as it strives to make sense of the raw information. One of the functions that occur within the brain as it interprets what the eyes see is assessing the relative distances of objects from the viewer, and the depth dimension of those objects.

Although the term "3D" is ubiquitously used, the presentation of dual 2D images is distinctly different from displaying an image in three full dimensions. The most notable difference is that, in the case of "3D" displays, the observer's head and eye movement do not change the information received about the 3-dimensional objects being viewed. Holographic displays and volumetric display do not have this limitation. Just as it is not possible to recreate a full 3-dimensional sound field with just two stereophonic speakers, it is an overstatement to call dual 2D images "3D". The accurate term "stereoscopic" is more cumbersome than the common misnomer "3D", which has been entrenched by many decades of unquestioned misuse. Although most stereoscopic displays do not qualify as real 3D display, all real 3D displays are also stereoscopic displays because they meet the lower criteria also.

The arrangement of the images created in with the nodes here, is left/right (side by side – short: SBS) with full width-resolution.

### 9.2 Depth Map

Depth information is taken from an AI model, generating depth map for a 2D image:



---

<sup>5</sup> Extract from Wikipedia

The quality of the depth map is important for correct depth feeling and improves with model quality. However, large model require significant longer time to process and in dynamic videos it is of less importance. Since videos need to be processed frame by frame it is usually sufficient and more efficient to use a small depth map generation model.

We apply some blurring to the depth map image to reduce artifacts. Can be changed by keys SBS\_DEPTH\_BLUR\_RADIUS\_VIDEO and SBS\_DEPTH\_BLUR\_RADIUS\_IMAGE in config.ini.

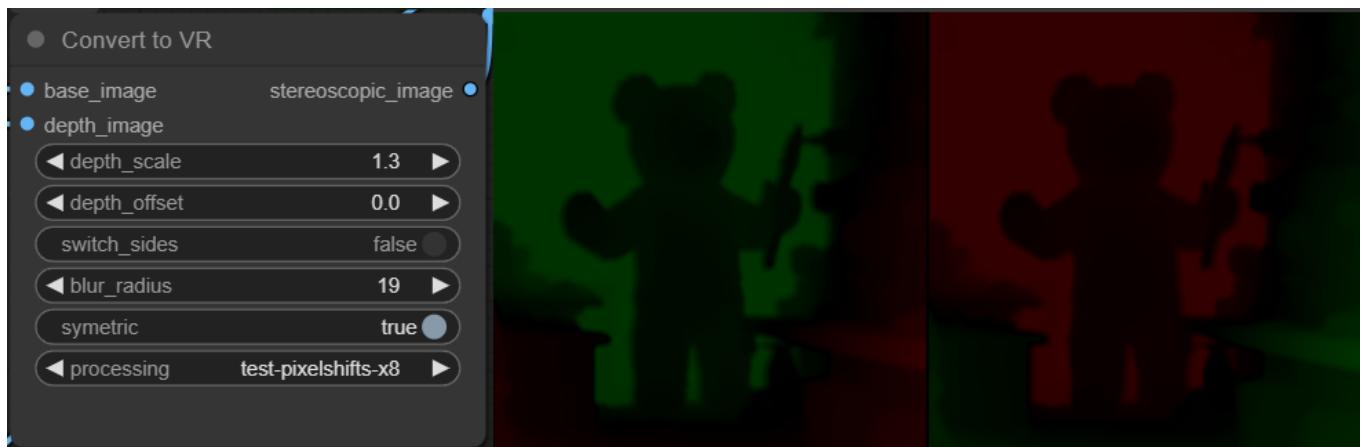
Depth model comes over ComfyUI custom nodes from `comfyui_controlnet_aux`. The model can be configured in config.ini by the key DEPTH\_MODEL\_CKPT. It defaults to `depthAnything_v2_vitb.pth`. The processing time of this model is about the same as the SBS converting algorithm. The giant variant must be installed manually.

### 9.3 SBS Conversion Algorithm

Left and right side of the image are generating by shifting pixels of the original image to the left or right, depending their value in the depth map. This tricks the eyes to let them show you the 3D effect. However, you can of course not see around objects, so at value “jumps” in the depth map, it can generate irritations to the eye. To overcome this situation, the depth “effect” scale should not be too big, it should be divided on both images equally and the resolution should be high.

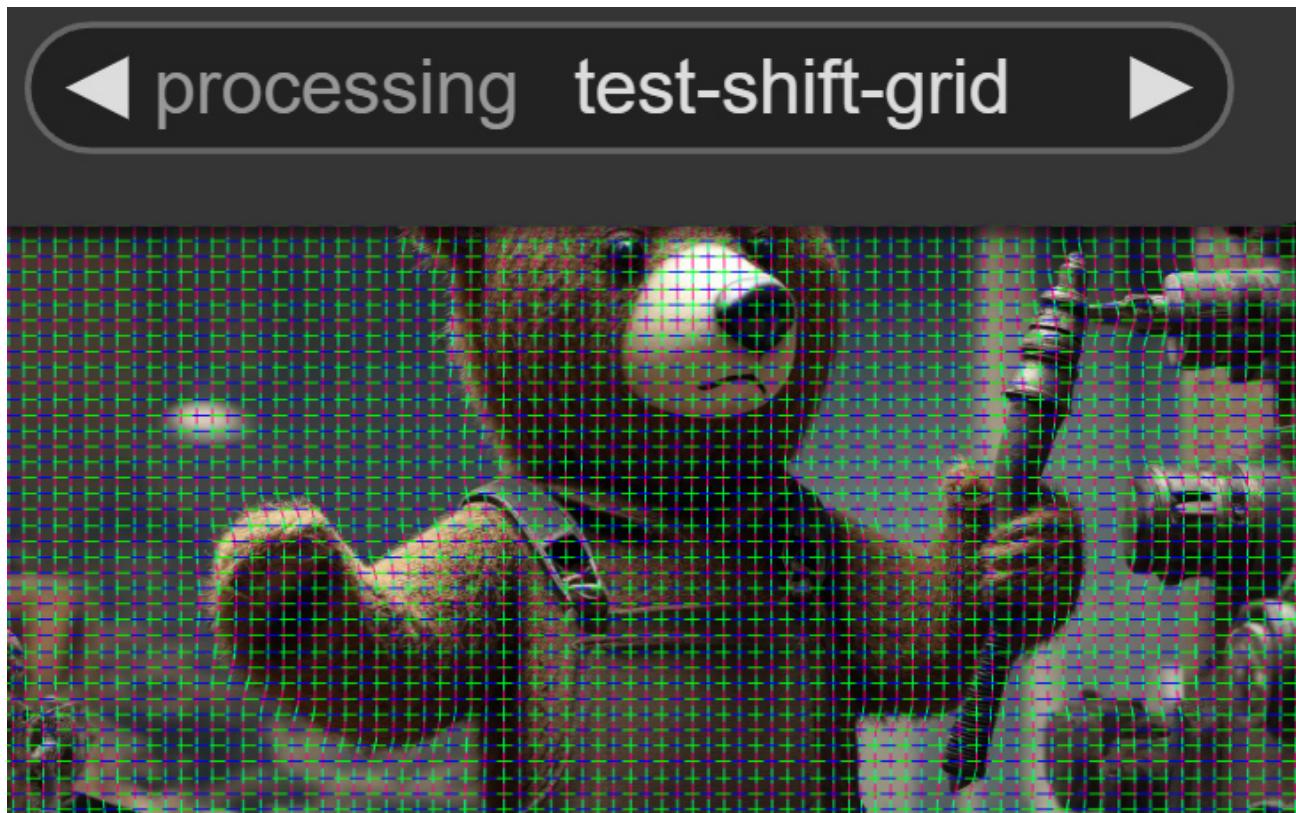
The following illustrations are made with an analysis workflow you can find under `custom_nodes/comfyui_stereoscopic/examples/workflows/analysis/I2I_SBS_Depth_Analysis.json`

So first the shift information is calculated, using the parameter `depth_scale` and `depth_offset`:



In config.ini they can be changed by the keys SBS\_DEPTH\_SCALE (default 1.0) and SBS\_DEPTH\_OFFSET (default 0.0). These values are normalized, half of the shift is to the back, the other half to the front and depth scale is balanced out to some natural feeling value. Higher depth\_scale makes the depth effect stronger, and higher depth\_offset brings the image more to the front. Symmetric is set to true for quality in my workflow; switch\_sides can't be adjusted though configuration, but for cross-eye fans there is a task called swapsides (see chapter 7.1.10).

Next the shift offsets are applied to image. You can see with another processing mode how the shift is applied to a grid laid over the image:



Finally we black-out some pixels at left and right side of the image to hide pixel artifacts generated due to lack of information in this area. To keep this area relative low, the resolution should be as high as possible.

The final result is a side-by-side image:



# 10 Advanced Features

## 10.1 Topaz Video AI

Using this software for upscaling and interpolation instead of using ComfyUI models and workflows archives much better results; it is at least 10 times faster and you can perfectly adjust parameters in a preview mode.

Currently tested with v6.0.4, models: prob-4, iris-2, chf-3

Topaz Video AI requires a valid login, that need to be renewed online after some time. The login procedure is not covered by this VR we are. Future versions of Topaz Video AI will not offer CLI support; Hence, new customers need to ensure they can still get a version of Topaz Video AI that supports it.

Known Bugs:

- Sometimes the ffmpeg task runs infinite. You can detect this when time stamp has no progress. In this case, kill the process with the TaskManger.

```
frame= 31 fps=0.0 q=37.0 size= 1kib time=00:00:00.64 bitrate= 14.7kbits/s speed=0.000739x
```



## 11 Troubleshooting

Problems or questions? Contact the Discord Support! Activation Link:  
<https://discord.gg/ZegT6Cc8FG>

- If an error occurs, the daemon.bat window may close or the error not invisible. Please start daemon.sh in a bash shell instead, to see the error. For just the user interface the script is gui/restart.sh. If there are errors you can report them on discord.

Some tips:

- For long videos: Use task *split-1m* first, at the end use stage *concat*.
- Some tasks may fail randomly. Try again.



## 12 Acknowledgements

The software was developed by me, Fortuna, with lot of help from lablunoshka, who is responsible for the insane speed of the SBS algorithm. We did this in the hope, that we soon see more VR content, from AI here or classic somewhere else.

Additional help come from some artists, namely (in alphabetic order): Robotka, schmede, VisionaryAI\_Studio and z1000k. You will see the result of this tool in the many examples in the gallery on Civitai. Please visit and rate their original work (links included in posts), instead of the stereoscopic variant. I want to say "thank you!" to them for the opportunity to propagate the potential of this tool through their artwork.

# Appendix: Default Workflow Pipeline

Best practice: Put your files at the arrows.

