industrial 3D robotics

Phobos operator manual

Revision date: Feb 2019

Contact Details

Correspondence Address:	I3D Robotics Ltd Pipers Business Centre 220 Vale Road Tonbridge Kent TN9 1SP UK
Email:	info@i3drobotics.com
Phone:	+44 (0)1732 373020 (09:00 – 17:00 UK)
Fax:	+44 (0)1732 373001

Revision date: Feb 2019

About this Manual

Who is this for?

This manual is intended for the operator of the I3D Robotics Phobos stereo camera. It will help the operator understand how to get the most out of this new product.

What's in it?

This manual provides the operator with instructions on how to unpack, install and operate the camera.

Specification	Contains an overview of the camera and its specifications.
Installation	Instructions on how to unpack and setup your system.
How do I use it?	How to get the most out of your stereo system
Troubleshooting	What to do if there's a problem

Warranty

This product comes with a 1-year warranty covering parts, labour and shipping needed to repair manufacturing defects that occur during the warranty period. Shipping costs are limited to shipping to and from customers during the warranty period. The warranty will be void if the system is handled inappropriately or if the housing has been opened.

Disclaimer

THE INFORMATION CONTAINED HEREIN IS PROVIDED "AS IS" WITHOUT WARRANTY, CONDITION OR REPRESENTATION OF ANY KIND, EITHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING BUT NOT LIMITED TO, ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL I3D ROBOTICS BE LIABLE FOR ANY LOSS OR DAMAGE, WHETHER DIRECT, INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR OTHERWISE HOWSOEVER CAUSED WHETHER ARISING IN CONTRACT TORT OR OTHERWISE, ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE INFORMATION PROVIDED HEREIN.

DEIMOS CONTAINS A CLASS 1 LASER DEVICE – YOU MUST FOLLOW GOOD LASER SAFETY PRACTICE WHEN USING THE PRODUCT. NEVER STARE INTO LASER APERTURES.

Revision date: Feb 2019

Revision date: Feb 2019

Contents

Revision date: Feb 2019

1 Introduction

Phobos is a high-resolution stereo vision system that uses 5 Megapixel cameras and can operate at 30 fps over a USB3.0 interface. The cameras use a global shutter and are suitable for imaging moving targets.

Some of the key features are detailed below:

Interface	USB3.0 (5 Gbps)
Camera resolution (px)	2,448×2,048 (5 MP)
Pixel Size	3.45 µm x 3.45 µm
Bit depth	Monochrome 8/12-bit
Sensor (x 2)	Sony IMX250LLR (2/3”)
Framerate	30 fps*
Focal length	8 mm
Focus	0.8 m to 1.25m**
Shutter type	Global
Synchronisation	Hardware triggered
Exposure range	20 µs to 4 s
Software compatibility	Windows 7+, Linux, OpenCV, ROS ROS

*Limited by USB3.0. This can be increased if USB3.1 is used. Acquisition framerate reduced to 14fps if 12 bit bit-depth is used.

** The Phobos stereo systems are factory-focused for an optimum measurement at 1m. Please contact us if you require a different measurement range as the focus range may be adjusted.

Revision date: Feb 2019

1.1 System requirements

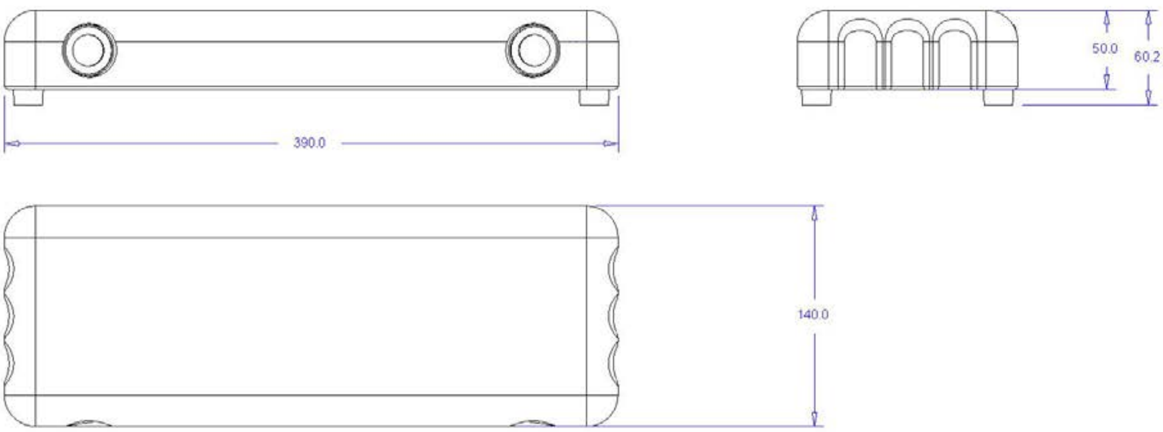
The table below outlines the minimal requirements of the connected computer running the user software.

Processor	Intel i3/5/7 or similar
RAM	4GB
Hard disk space	500MB
Graphics card	CUDA compatible card. Geforce-gtx-1060 or better recommended
Connectivity	USB3

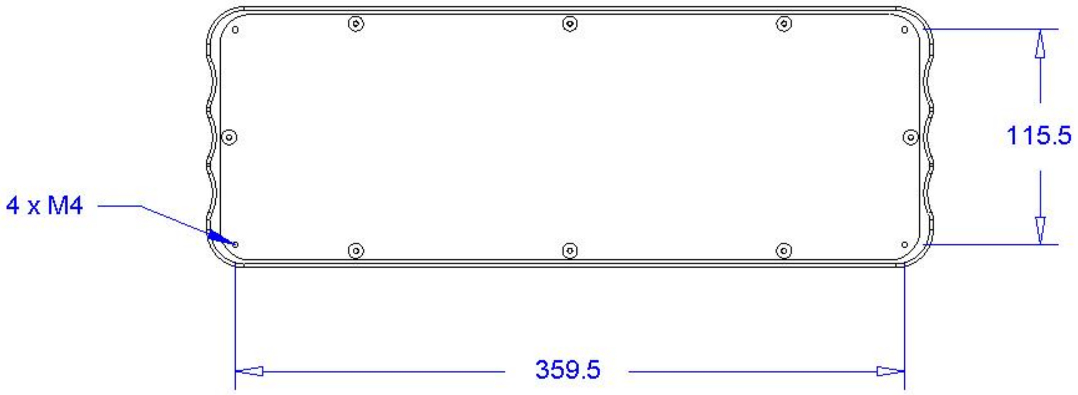
1.2 Mechanical and Electrical Information

Dimensions	390 x 140 x 50 mm
åManufacturing tolerance	±1 mm
Weight without cable	2.83kg
Mounting	2 x M4
Temperature range	5 – 50 °C
Power requirements	5V /3A

Dimensional drawing, all dimensions in millimetres. This drawing is reproduced here for reference, please contact us for a higher resolution copy or CAD models.



Revision date: Feb 2019



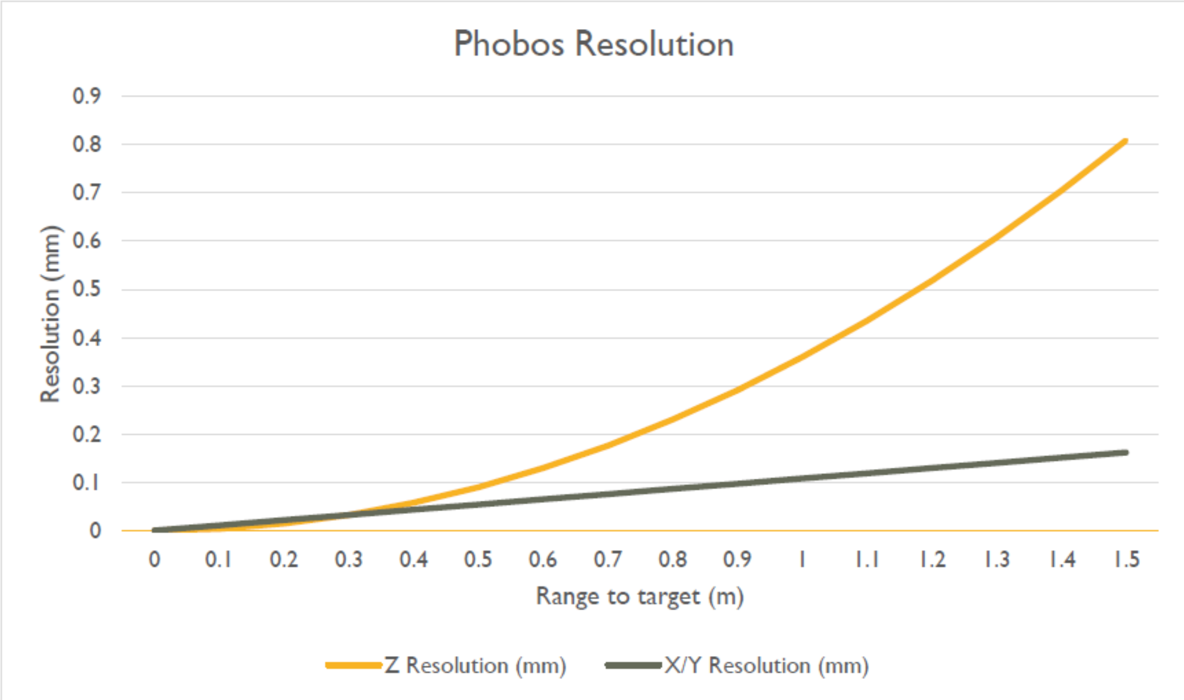
1.3 Certification

This product conforms to 2012/19/EU (WEEE), 2011/65/EU (RoHS) and 2014/30/EU (EMC Compatibility).

2 Stereo specification

Z axis indicates depth, perpendicular to the camera sensor plane.

Baseline	300 mm
Depth resolution at 1 m ($\in_z = \frac{z^2}{fB} \in_d$), $\in_d = \mathbf{0.25}\ \boldsymbol{px}$	0.35 mm
X/Y resolution at 1 m	0.11 mm
Field of view at 1 m (H x V)	1.05 x 0.88 m (57.65° x 47.65°)



Given a point (X, Y, Z) which is located at (x, y) in the left image and a known disparity, d, along with the camera separation (baseline) b and focal length f:

$$x = \frac{fX}{Z}, y = \frac{fY}{Z}, Z = -\frac{bf}{d}$$

Stereo range resolution therefore determined by disparity, for a fixed b and f, as shown in above. These are representative of the performance of Deimos and should be regarded as theoretical. Stereo range accuracy (σZ) is (quadratically) distance dependent, for a fixed b and f:

$$\sigma Z = \frac{Z^2}{bf}\sigma d$$

Where σd is the uncertainty in disparity measurement, nominally a quarter pixel. This is dependent on the choice of stereo matcher.

3 Installation

3.1 Unpacking and handling

As with any optical instrument, you should take care when handling the stereo camera. Take care not to touch the lenses if possible.

The enclosure is reasonably dust-proof, and will work outdoors, but it is not explicitly protected against water or particle ingress.

Once all the elements have been removed from the packaging you should check the contents.

3.2 What should I have?

The contents of the package you have just purchased will depend on the options you selected when purchasing.

The basic system comprises the following items:

- Phobos stereo unit
- USB3 Cable
- USB key containing software, documentation and factory calibration
- 5V Power Supply

4 Operation

4.1 Using the Stereo Vision Toolkit

We provide an open source GUI tool to experiment with your camera on Windows. The tool can be used to acquire images, perform stereo image matching and view live 3D point clouds. For most applications, we would expect that customers will integrate the camera using their own software (e.g. using the OpenCV library), but our toolkit is designed for rapid application prototyping.

The software installer is provided on a USB stick, or may be downloaded from our Github repository (https://github.com/i3drobotics/stereo-vision-toolkit) either as a binary, or as source which you can compile yourself. If you wish to compile from source, we provide library dependencies built using Visual Studio 2015 x64 in the repository.

4.2 Installation

An installer for the toolkit software is on the provided USB key. You can find the latest release on the repository (link above).

You will have been provided with a factory stereo calibration file on a USB stick with your camera. Place this in your installation folder, in the 'params' folder as the software will look there by default. Alternatively, you can specify the location in-software and the path will be saved in the application's configuration file.

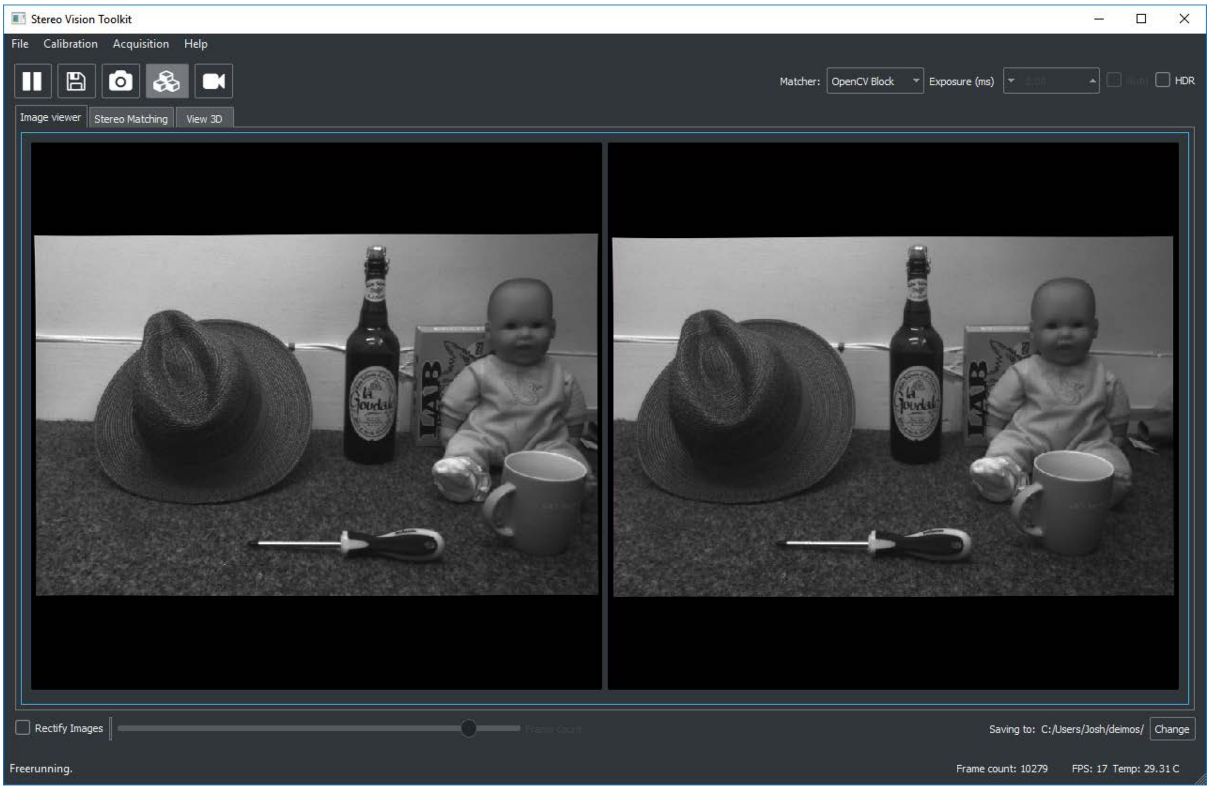
Notes:

- You may need administrator privileges to copy these files if you installed the software to a protected folder (e.g. Program Files).
- The toolkit will store its configuration in the Windows registry at: Computer\HKEY_CURRENT_USER\Software\I3Dr\Stereo Vision Toolkit
-

4.3 Acquiring images

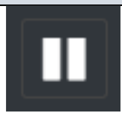
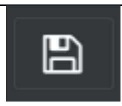
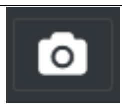
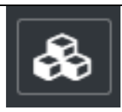
First, make sure your Phobos unit is plugged into a USB3 port to achieve the required framerate.

Open the software, you will be presented with the stream from the two cameras. The acquisition tab is shown below:



The different features of the toolkit are arranged by tab. The first tab shows the stream from each camera. You can check the frame rate in the status bar at the bottom, as well as the camera's reported temperature from an onboard sensor. This is calculated using an assumed ambient temperature of 20 C. Also, in the status bar, you can see the frame count[1] and the current output directory.

The toolbar provides you with the following options:

Command	Button	Description
Play/Pause		Start or stop the camera image stream
Save		Save the current frame to a user-specified folder
Snapshot		Grab a single frame and then pause
Enable matching		Toggle stereo matching
Record video		Start or stop recording a stereo video

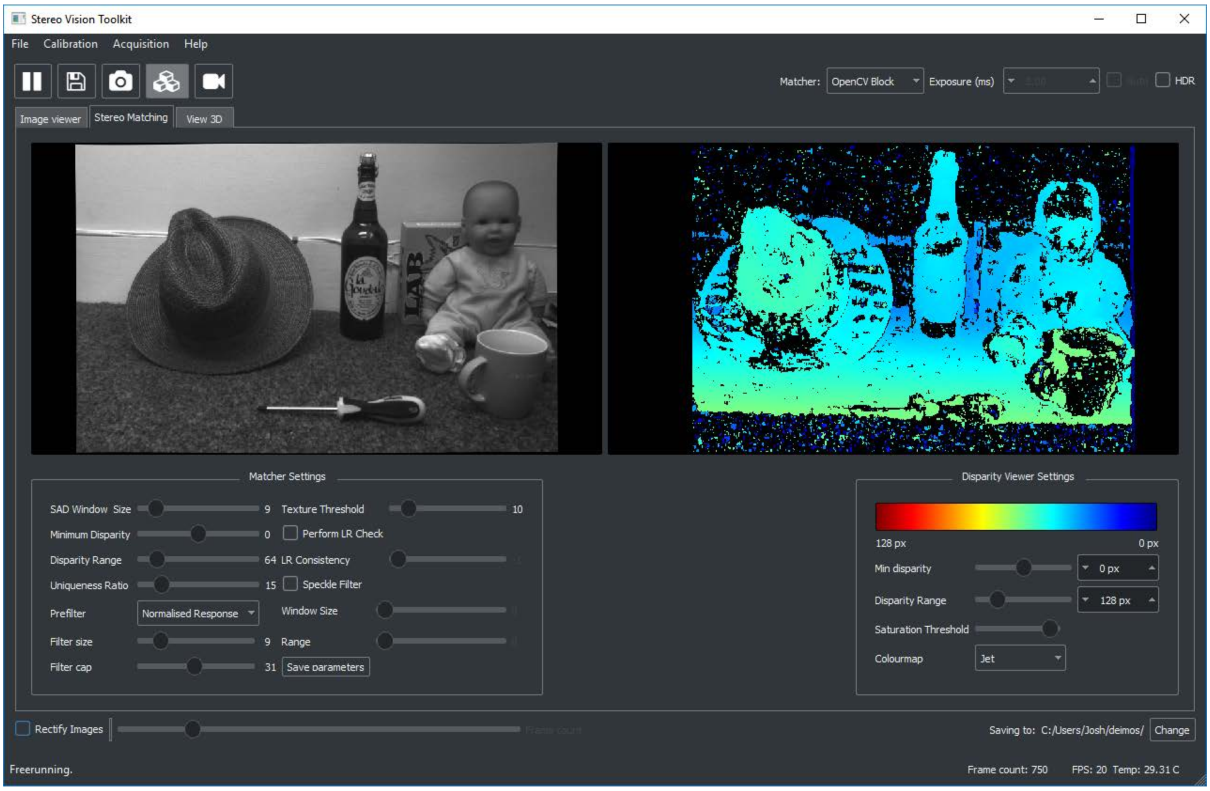
[1] This can be helpful as an indicator that the camera is still capturing images.

At the top right, you can specify which matcher to use (OpenCV Block, OpenCV SGBM or i3Dr's SGM), the camera's exposure time in ms (or auto-expose) and whether to toggle HDR mode. HDR mode is useful for suppressing highlights in the scene, and does not affect the maximum framerate, especially in conjunction with auto-expose mode.

4.4 Matching images

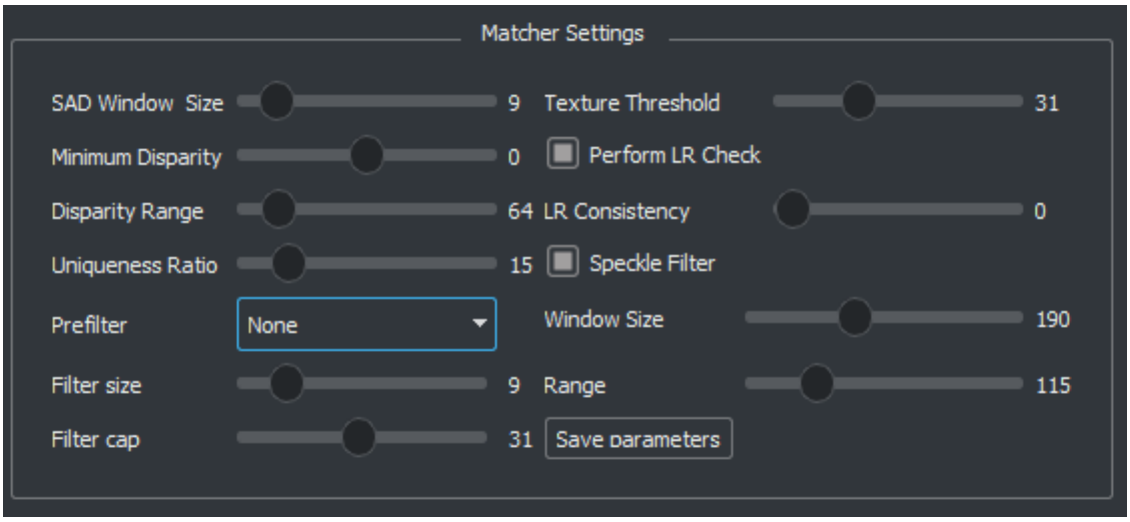
Once you have loaded in the correct calibration files for your camera (See Section 4.6 for details on how to calibrate), you can enable image rectification and start matching. Alternatively, if you are loading a video that's already been rectified, then you can enable matching regardless.

The matching window has three sections: the view from the left camera, the disparity map and the parameters for the selected stereo matcher. Parameter changes will update in real time, allowing you to experiment to find out what works best for your scene.



Two matchers are provided currently, both from OpenCV. The first is a block matcher which local area matching[2]. This is a fast and simple method which can give good results, particularly if using the projector. However, it can struggle with repetitive objects. The second algorithm is the OpenCV implementation of Semi-Global Matching which is a more sophisticated algorithm and runs quite a lot slower than block matching.

[2] It compares a small patch in the left image and attempts to locate its match in the right image using correlation.



There are several common parameters (further description may be found in OpenCV's documentation here)

Window size – sets the size of the comparison region. Bigger values will produce denser disparity maps at the expense of fine detail and edge sharpness.

Minimum disparity – set the minimum disparity to search from.

Disparity range – set the number of disparities to search over.

Uniqueness ratio – sets how confident the algorithm must be before a match is allowed. Set this to a high value to improve robustness at the expense of disparity map density.

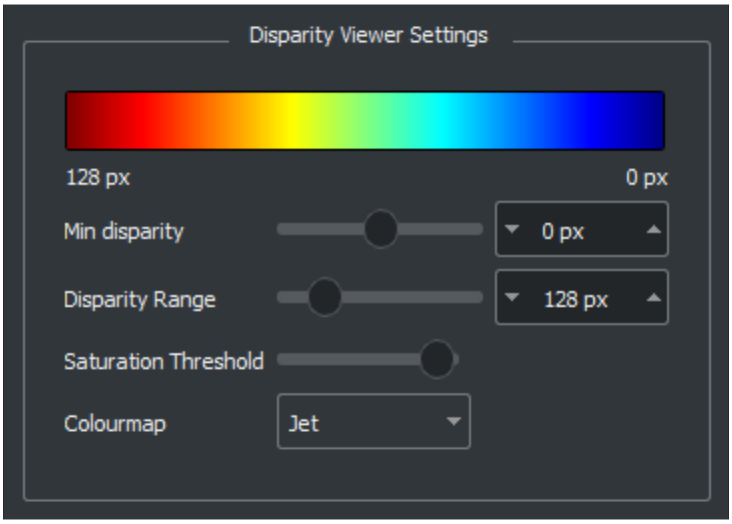
LR Check – enable an approximate left-right consistency check, which can reduce noise in the disparity map.

Texture threshold – minimum image texture to allow matching, can avoid spurious matches in featureless regions.

Speckle filter – a post-processing step to remove small outliers in the disparity map. Generally setting both range and window size to be high works well.

Block matching also has the option to pre-filter the image which can improve noise in the disparity map.

When you are happy with the parameters you have chosen, you can opt to save them as defaults for the next time you run the application.



On the right, you can control the visualisation settings for the disparity map including the disparity minimum and range (the same as for the matcher), the choice of colourmap and a filter to filter pixels that are brighter than a certain amount. Saturation filtering can be useful if very bright lights are observed by the cameras.

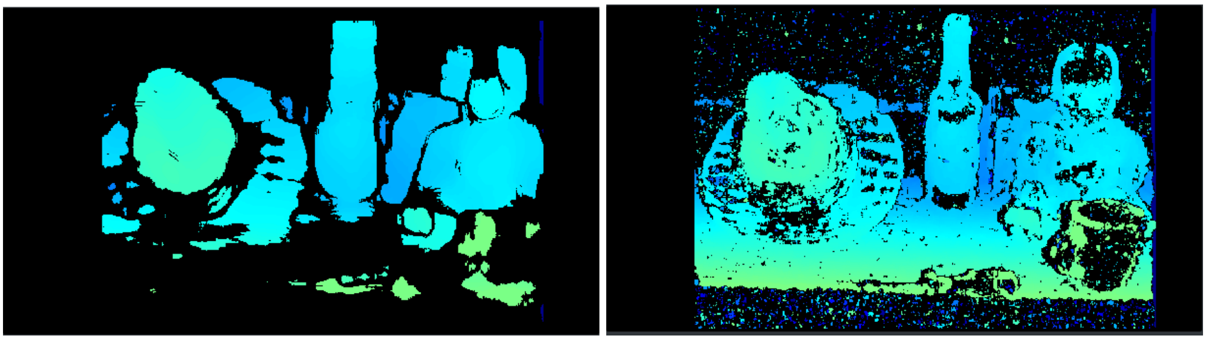


Figure 1 Left: disparity map with a SAD window size of 30 px, Right: disparity map with a SAD window size of 9 px

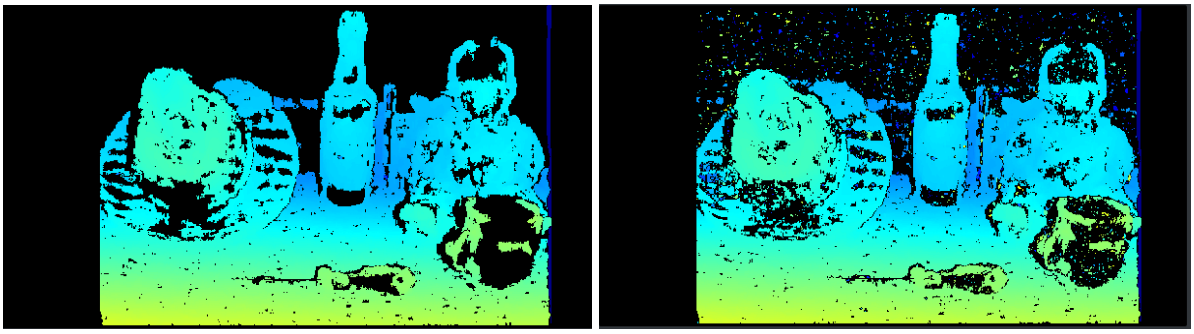


Figure 2 Left: Speckle filtering enabled, Right: no speckle filtering

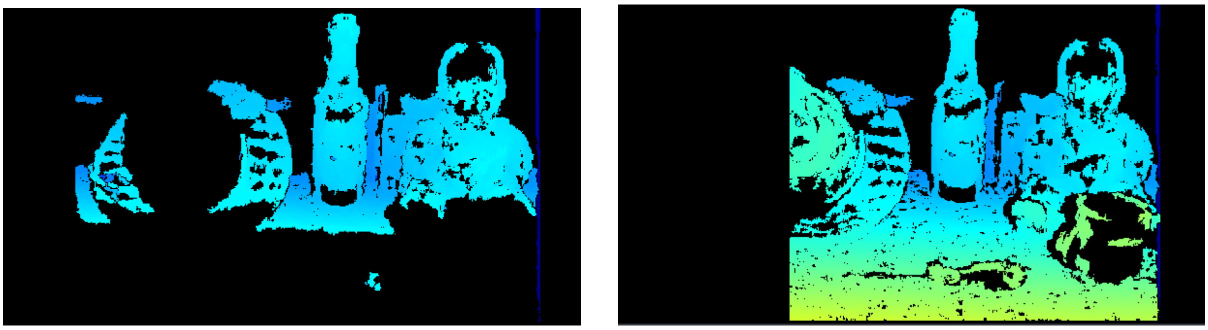


Figure 3 Left: A small disparity range of 32 px does not match the whole image. Right: A higher range of 192 px matches more of the image closer to the camera, but also reduces the region of the image that can be matched.

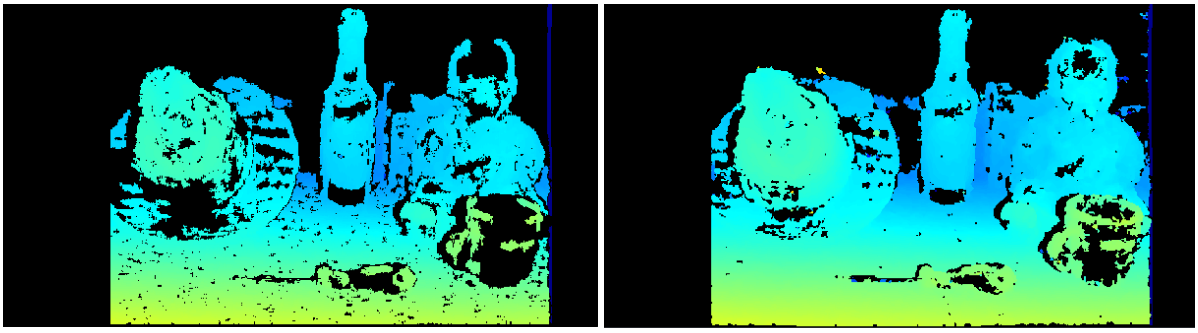
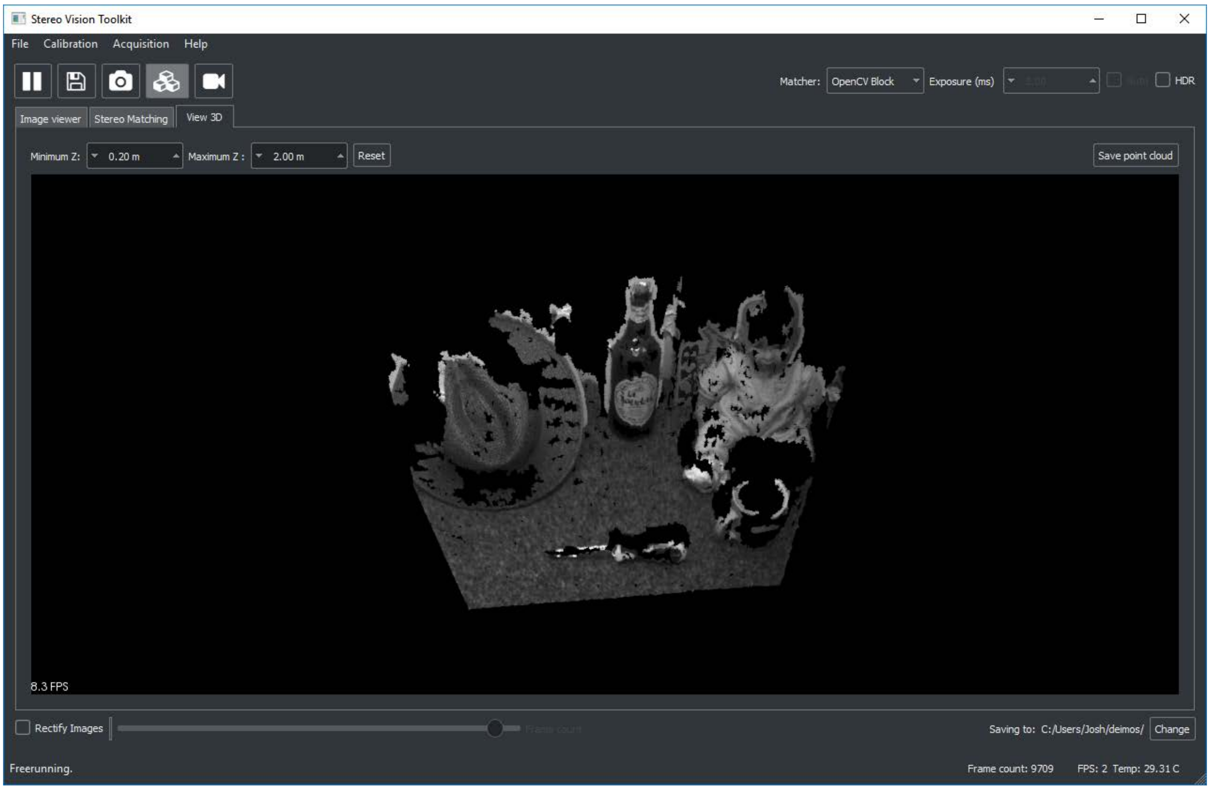


Figure 4 Left: OpenCV Block Matcher Right: OpenCV SGB Matcher

For most applications you will probably need to tweak the matcher settings to get the best reconstruction for a specific scene, and depending on the desired trade-off between point cloud sparsity and robustness.

4.5 Viewing point clouds

If the camera is properly calibrated, you can switch to the third tab to see the disparity map projected into 3D space. The point cloud is displayed with the image intensity overlaid on top of it.



You can zoom with the mouse wheel and rotate with the left mouse button. Holding shift and clicking pans the scene.

Set the min/max Z distance to limit the range of points shown.

You can save the current point cloud as a PLY file for viewing in other software, such as the freely available Meshlab or CloudCompare.

Click the reset button to refresh the view. Occasionally the point cloud will not appear immediately and you may need to shift the view slightly to re-render it.

4.6 Calibration

Calibration is required to reconstruct 3D images using Deimos, or any stereo camera. The purpose of camera calibration is to obtain the intrinsic parameters of each camera, which include the focal length of the lens, the intersection between the lens optical axis and the sensor plane (camera centre) and the characteristics of the lens' distortion. This is sufficient to 'undistort' a single camera – a simple way to think about it is that straight lines in the world become straight in the undistorted image. For stereo systems, an extrinsic calibration is also needed which describes the relative positions of the cameras with respect to each other. Combined with the intrinsic calibrations for each camera, a rectification transform is calculated which warps the raw images prior to stereo matching.

Your camera is provided with a (tested) factory calibration which should remain stable in transit. However, you may wish to perform your own calibration and sooner or later you will probably need to. If you adjust the focus of the lenses yourself, or if for some reason you find that the images are failing to match (even with the projector on) then you can try a re-calibration.

First, print off a calibration target: https://docs.opencv.org/2.4/_downloads/pattern.png

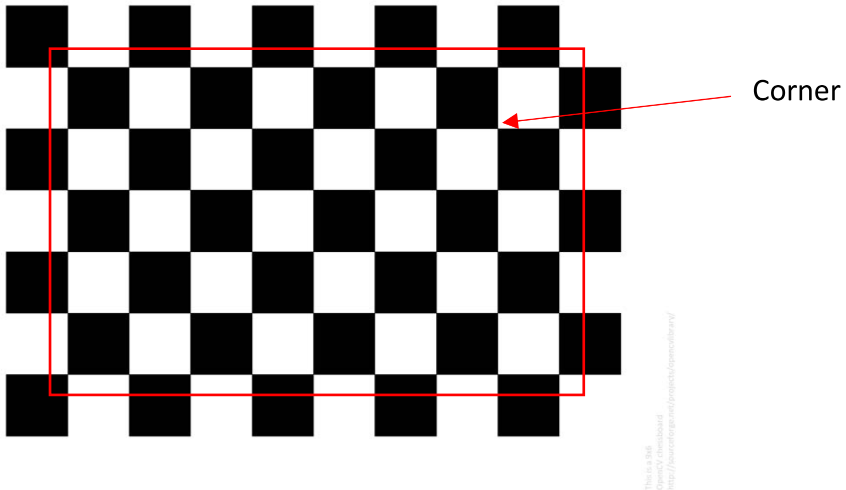


Figure 5 A standard 'chessboard' calibration target. We refer to the calibration points as corners, the points where the square vertices meet. The red box highlights the 'active' area of the pattern, which should be visible in all images.

You should print with as high a resolution as your printer will allow. An A4 target is the minimum recommended size, you may have better results with an A3 or even A2 target. Ensure that the aspect ratio of the print is fixed, if you have rectangles rather than squares, the calibration will be poor. If using a poster printer, be aware that glossy finishes usually suffer from specular reflections from room lighting.

Fix the printed calibration to a stiff board or surface, such as a sheet of wood or aluminium. The surface should be as close to planar as possible. Avoid taping the edges of the target, as over time paper tends to warp and the target will no longer be flat. A better solution is to glue the target to the board.

Measure the width of one of the calibration squares as accurately as you can – ideally using the software you printed the target with, if possible (print 1:1). The calibration algorithm attempts to locate 'corners' – points where the corners of the squares meet (see Figure above).

Next, specify a convenient save directory and capture some stereo image pairs of the target in different positions and orientations. It is important that you include images of the pattern tilted, and at various points and scales in the field of view. If your images are too similar, your calibration will be poor. This is important to accurately determine the distortion parameters for the lenses.

A typical selection of orientations might be:

1. Face on, close to camera
2. Face on, far from camera
3. Left image edge, tilted at various angles
4. Right image edge, tilted at various angles
5. Top image edge, tilted at various angles
6. Bottom image edge, tilted at various angles

Some example images (left only) are shown below:

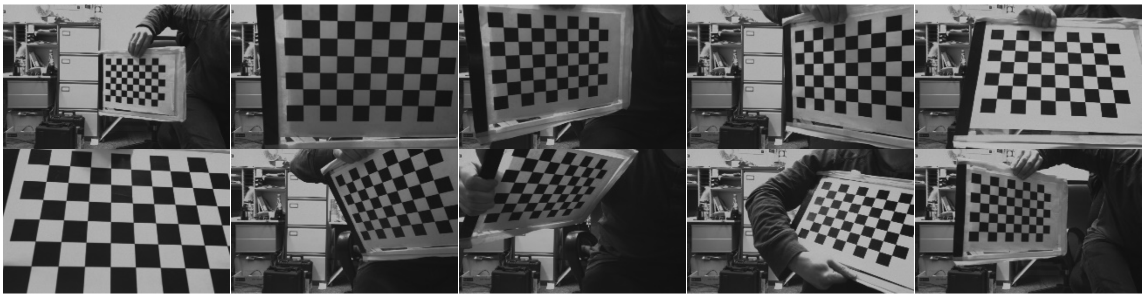
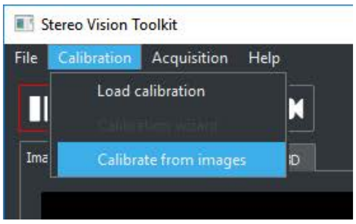


Figure 6 An example set of calibration images, using an A4 target printed on a board.

Capturing 10 calibration pairs should be enough for many cases. Make sure the pattern is fully visible in both cameras. Once you're done, open the calibrate from images tool:



Follow the instructions in the dialog window, providing the location of the left and right calibration images. The default image file mask should work. Make sure you set the correct pattern size (the image above is an 6 x 9) and the correct square size in millimetres.

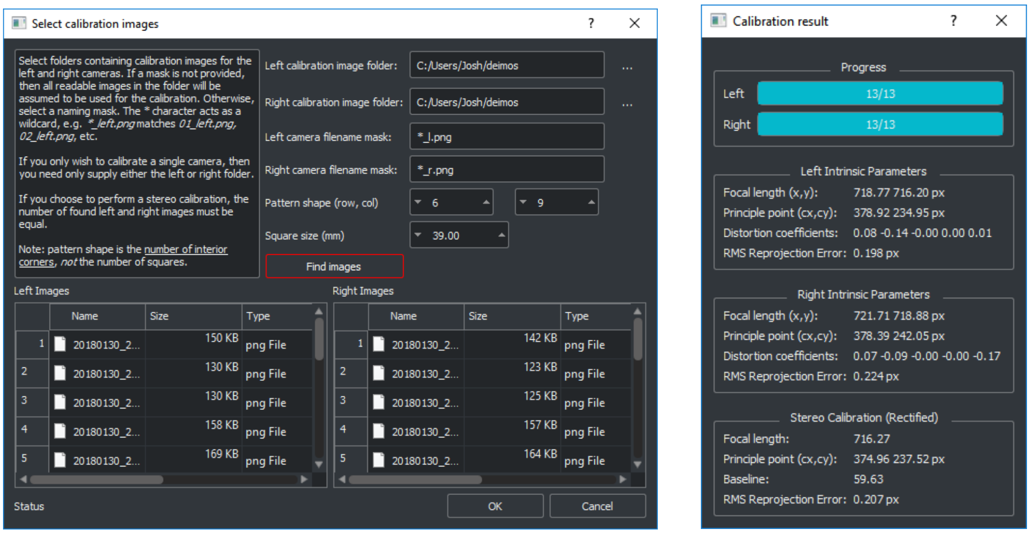


Figure 7 Left: the calibrate-from-images tool. Right: the calibration result using the calibration images in Figure 5. Note the good stereo RMS reprojection error of 0.2 px.

Click "Find images" and ensure that the images you just captured appear in the table. There should be an equal number of left and right images.

Click OK to perform the calibration. The calibration will take a few seconds to complete, depending on how many images you used. Once done, you will see a dialog with the result. A good calibration error (RMS reprojection error) is less than 0.25 px.

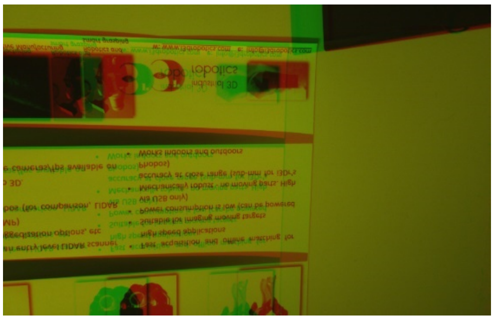
Finally, load your calibration files by selecting the folder that the calibration was saved to (currently the application directory). You can move the files somewhere else if you wish to, e.g. for backup or if using multiple Deimos cameras. Check the calibration works by seeing if you get sensible stereo matching results.

5 Using external libraries

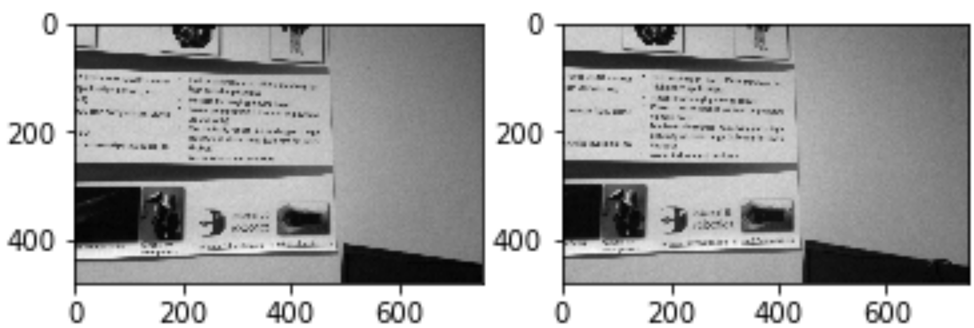
5.1 OpenCV

You can use the Deimos camera in OpenCV (C++/Python). Simply load the camera as a VideoCapture object and set the resolution to 752 x 480. Frames are returned as a 3-channel colour images. The last two channels correspond to the left and right images respectively.

```
from matplotlib.pyplot import *

import cv2

cap = cv2.VideoCapture(0)

cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

cap.set(cv2.CAP_PROP_FRAME_WIDTH, 752)

image = cap.read()[1]

image = np.flipud(image)

imshow(image)
```



```
f = figure()

subplot(121)

imshow(image[:,:,1], cmap='gray')

subplot(122)

imshow(image[:,:,2], cmap='gray')
```



Note that manual exposure control is not possible directly using OpenCV. This is controlled using a HID interface to the camera. Example code for controlling the camera in C++ can be found in the stereo vision toolkit.

Revision date: Feb 2019

5.2 Using ROS

A compatible ROS node for the camera may be found here. This node also provides access to the camera's onboard IMU as a topic and allows you to control the camera exposure.

Since the camera enumerates as a UVC compatible device, you should be able to use Deimos in most imaging libraries with generic capture interfaces.

6 Troubleshooting

6.1 Framerate is slower than expected

Check that you are not performing any graphics-heavy processing on the same machine. The stereo vision toolkit automatically performs rectification using your GPU (if available). Similarly, if you are doing any heavy computation work, this may slow down stereo matching.

6.2 I only get 30fps, not 60fps

This is typically because the camera is operating in USB2 mode. Please check that you are using a USB3 port, and that it has sufficient bandwidth. Alternatively try using a different USB cable.

6.3 Image matching is poor

It's possible you need to recalibrate the system. First, try to image a target that is amenable to matching, with lots of features. A blank wall with the projector turned on is a good option. If you still do not get good match results, then perform a recalibration and try again. Check that the rectification appears to be sensible – straight lines in the world should be straight in the images.

6.4 USB connector is not secure

This is a common problem USB3 Micro-B connectors over time, particularly if they are removed and re-inserted frequently. We provide the Deimos system with tapped holes, usable with standard USB3 vision cables (with screw locks). You may find that another USB3 cable provides better latching.

6.5 I'm having problems with the software

Please visit the Github repository and post an issue (preferred), or contact us for assistance. Remember

The camera does not need additional drivers.

6.6 I'm getting high calibration errors (> 0.3 px)

- Make sure that you are in a brightly lit area, and lower the exposure time, to prevent motion blur
- Use HDR mode
- Capture more calibration images, making sure that the board is present at different distances and with a large variety of orientations relative to the sensor

- Check that the pattern size you provided is correct

6.7 I can't see any 3D output

Sometimes you may need to move the view slightly (using your mouse) to re-render the point cloud. You need a valid calibration to project to 3D, as the camera projection matrix must be known.

6.8 Where can I find out more about stereo imaging?

The canonical reference textbook for stereo imaging geometry is Hartley and Zisserman, *Multiple View Geometry in Computer Vision* (OUP), however there is little information on matching. Unfortunately there are few good books on stereo, in terms of a good algorithm comparison, so it's usually better to look at the relevant papers.

You can look at Scharstein and Szeliski's stereo taxonomy paper (https://doi.org/10.1109/SMBV.2001.988771) or the OpenCV calib3d documentation (https://docs.opencv.org/2.4/modules/calib3d/doc/calib3d.html). For information on the calibration routine used, take a look at Zhang's paper (https://www.microsoft.com/en-us/research/publication/a-flexible-new-technique-for-camera-calibration/).

You can also look at the Middlebury and KITTI stereo benchmarks for a more comprehensive algorithm comparison, though note that the test data has existed for a while and there is likely to be some bias towards matchers which work well on that data. For example, the top KITTI performer is a deep neural net trained on driving imagery, so it's not clear how well it would transfer to other domains.