OMG-V2.0 B - 13/01/20

USER MANUAL

1 Index

2 Introduction

Omega (head) provides an Application Programming Interface (API) allowing communication with a computer or control unit (CU). The head, which refers to the stereoscopic camera, makes several data available on the API, such as the raw images (left and right), the disparity image and calibration information. This data allows the development of specific applications or products adapted to a wide range of contexts and markets.

The API allows a Microsoft Windows or a Linux based application to connect to and communicate with Omega. To get a full list of what the Omega API may offer, please refer to the Omega API Reference guide.

The software installed on the head is called firmware. The head is equipped with an FPGA, which computes the dispdarity image in real time.

There are two possible Omega versions: Standard and HD. Depending on your hardware version, the following bitstreams may be available:

- Census bitstream for Standard version
- SGM bitstream family for HD version

The bitstream is a sub-part of the firmware, related to the embedded FPGA. See the next section “Compatible bitstream” to get more information.

3 Compatible Bitstreams

3.1 Census Bitstream

The Census bitstream

- provides a disparity map based on a census algorithm https://en.wikipedia.org/wiki/Census_transform
- disparity map size is 496x250
- implements disparity filtering but not configurable
- with algorithms which are a full hardware implementation to provide
  - low latency
  - energy efficient solution

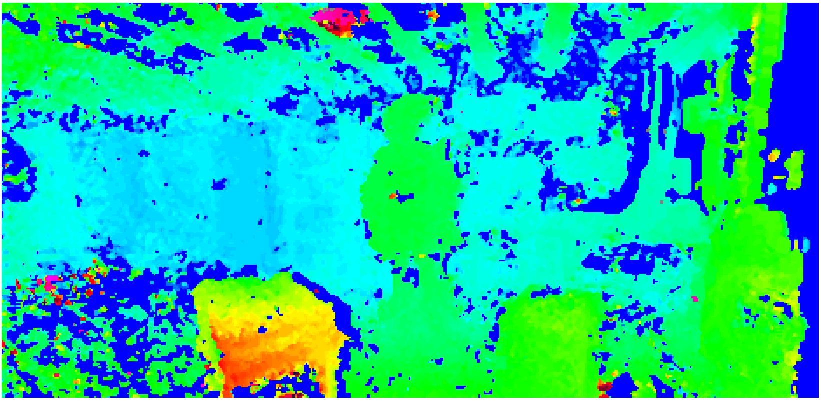


Figure 1: Disparity census

The census bitstream provides several filterings to enhance quality of the disparity map. The filters are:

- a left-right consistency check performs a left-right comparison to identify the mismatched regions which may probably contain erroneously labeled pixels.
- a median filter remove outliers in the disparity map
- a confidence check invalids pixels that belong to unmatchable regions..

3.2 SGM Bitstream Family

SGM bitstream family

- provides a disparity map based on a Semi-global matching algorithm https://en.wikipedia.org/wiki/Semi-global_matching
- allows a set of predefined sizes for disparity map (one bitstream per size)
  - 1024x512
  - 768x384
  - 512x256
  - 256x128
- implements disparity filtering
- allows to configure disparity filtering by using the API
- with algorithms which are a full hardware implementation to provide
  - low latency
  - energy efficient solution

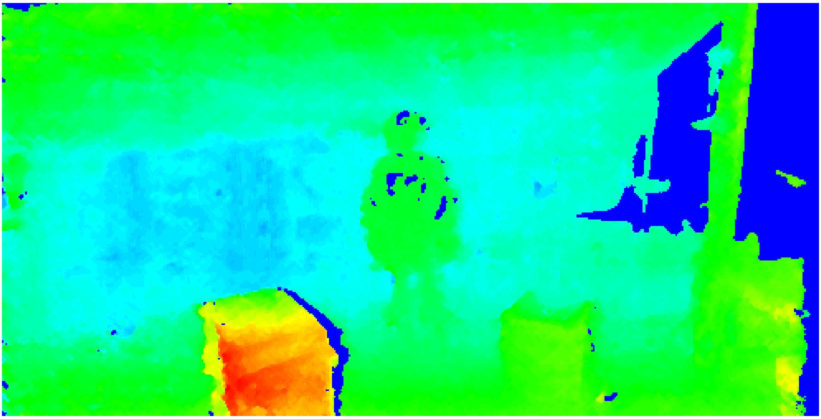


Figure 2: Disparity SGM

SGM bitstream family provides several filterings to enhance quality of the disparity map. The filters are:

- a left-right consistency check performs a left-right comparison to identify the mismatched regions which may probably contain erroneously labeled pixels,
- a median filter removes outliers in the disparity map,
- a confidence check invalids pixels that belong to unmatchable regions,
- an intensity threshold invalids pixels in over-exposed regions,
- a sub-pixel scaling configuration in order to increase the disparity size by interpolation. Disparity map quality increases with sub-pixel scaling since the disparity image compression is adapted to a working zone.

4 Supported OS

Omega API is compatible with the following operating systems:

- Windows x86_64
    - Windows 7
    - Windows 10
- Linux x86_64
    - Debian 9
    - Ubuntu 16.04
    - Ubuntu 18.04
- Embedded platform
    - Ubuntu 18.04 (Nvidia Xavier)
    - Ubuntu 18.04 (Nvidia Jedson)
    - Ubuntu 18.04 (Nvidia Nano)
    - Raspian 10 (Raspberry pi 4)
- Node ROS available for
    - Kinetic (Ubuntu 16.04)
    - Melodic (Ubuntu 18.04)

We only provide support and packaging for the operating systems listed above. Other systems are not guaranteed to work correctly, it may work without any additional effort or may need considerable efforts to configure.

5 Content

The camera is delivered with its API installer. When included in a kit, the camera is delivered with its accessories, which are a mounting support, a connection box, a 5m additional Gb ethernet cable and a power supply transformer.

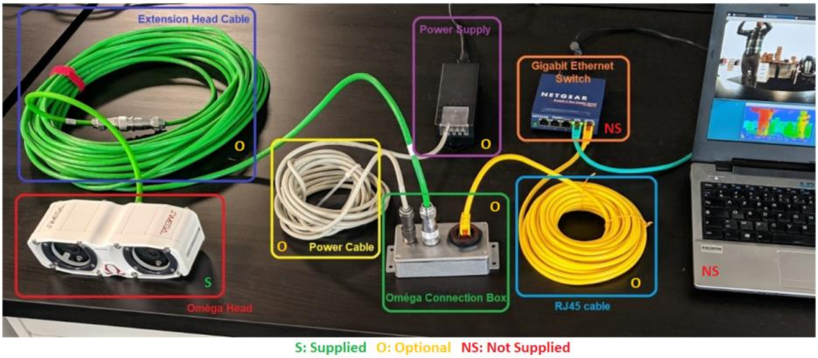
5.1 Omega API

The API installation package contains:

- **ASH Run-time libraries:** shared libraries to deploy with applications using the ASH API.
- **Source code of several example programs:** a “quick start” on how to use the ASH API and develop your own applications.
- **Documentation:** the ASH API reference guide and the present user manual.

In order to install the API on the CU, choose the file compatible with your Operating System (OS) and see installation section.

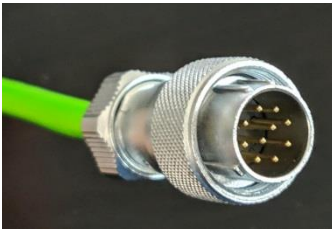
5.2 Omega setup



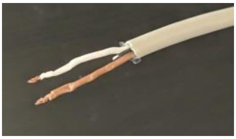
Not featured on the picture: Head Fixing Mount & API USB Key

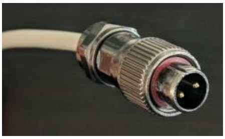
Figure 3: Omega Setup

Head cable to connection box
Connector: 8 Gigabit pins + 2 Power pins
Cable length: 3, 5, 10, 15 or 20m
(5m additional length is included in the kit)
Bending radius is superior or equal to 50mm



Connection cable box to Power supply
Cable length: 5m
White: negative pin (-)
Brown: positive pin (+)





RJ45 Gigabit cable (Yellow)
Max length: 20m (not supplied)

Omega connection box
This box splits the power & the RJ45 cables
Regulator:
IN: 9 - 36 UDC / **OUT**: 12 UDC / 1250 mA

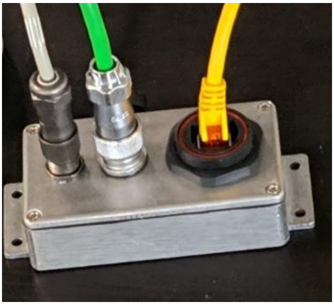


Figure 4: Omega connection

6 Network Setup

The ASH Omega has 3 network configurations:

- Avahi (factory configuration)
- DHCP
- Ip Static

The following configuration description is for the factory configuration (Avahi).

In general, the CU and the head must be in the same network (LAN), connected directly or via a Gigabit switch.
The CU must also have a Gigabit Ethernet interface in order to communicate with the head. The CU's firewall must be disabled.

The table below lists the network configurations that have been tested with the API. Any other setup is not guaranteed to work.

Setup	Network setup description	Windows	Linux
1	CU with a single enabled Gigabit Ethernet interface wired to a Gigabit switch. One head wired directly to the same switch using a Gigabit cable. The switch is connected to other peripherals (e.g. LAN of a company). The CU's firewall is disabled.	Not supported	Supported
2	CU with a single enabled Gigabit Ethernet interface wired to a Gigabit switch. One head wired directly to the same switch using a Gigabit cable and no other peripherals are connected. The CU's firewall is disabled.	Supported	Supported
3	CU with a single enabled Gigabit Ethernet interface directly connected to one head. The CU's firewall is disabled.	Supported	Supported
4	CU with a single enabled Gigabit Ethernet interface wired to a Gigabit switch. Multiple heads (up to 10) wired directly to the same switch using a Gigabit cable. The switch is connected to other peripherals (e.g. LAN of a company). The CU's firewall is disabled.	Not supported	Supported
5	CU with a single enabled Gigabit Ethernet interface wired to a Gigabit switch. Multiple heads (up to 10) wired directly to the same switch using a Gigabit cable and no other	Not supported	Supported

Setup	Network setup description	Windows	Linux
	peripherals are connected. The CU's firewall is disabled.		

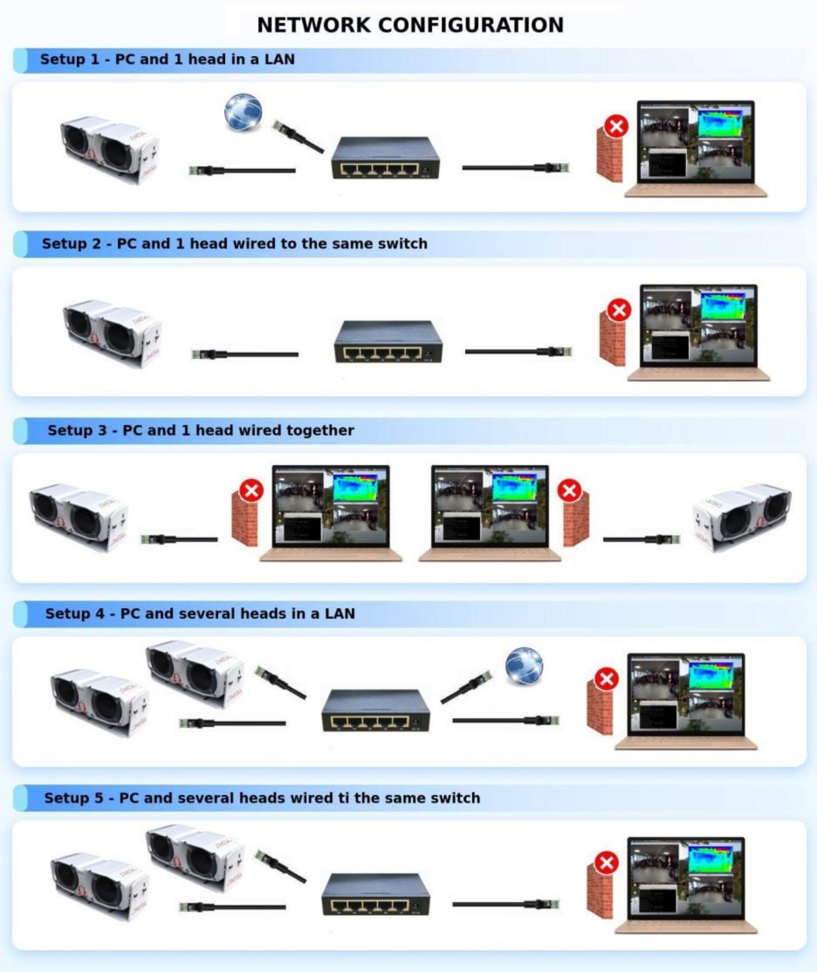

NETWORK CONFIGURATION
Setup 1 - PC and 1 head in a LAN
Setup 2 - PC and 1 head wired to the same switch
Setup 3 - PC and 1 head wired together
Setup 4 - PC and several heads in a LAN
Setup 5 - PC and several heads wired ti the same switch

Figure 5: Network setup

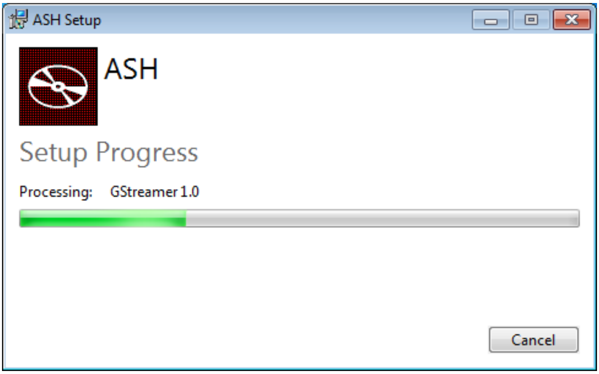
7 Installation

7.1 Prerequisites

- Disabled firewall;
- Uninstall any previous version of the ASH;
- Elevated privileges for package installation (super user or administrator).

7.2 Windows

Run ASH_WINDOWS-7-10_x86_64b_v2.0.X.exe for Windows version, then follow the instructions provided on the screen.



Processing: GStreamer 1.0

Figure 6: ASH Installer under Windows 7/10

The source code of the examples is installed in %ASH_PATH%\share\ash\examples

The binaries of the examples are installed in %ASH_PATH%\bin\ASH*.

NOTE

1. Always keep the last installer, don't move or delete it as it will be used by the package removal.
2. Perform a clean install: Always remove an installed package (using the last installer you kept) before installing a new version of the installer.

7.3 Linux

Open a terminal in the directory containing the package installer. Run the following commands:

```
$ sudo apt-get update
$ sudo dpkg -i <package>
$ sudo apt-get -f install  # To fix missing dependencies
$ sudo dpkg –i <package>
```

Replace <package> by the one that corresponds to your operating system and architecture (ex: ASH_UBUNTU-18-04_x86_64b_v2.0.X.deb)

The source code of the examples is installed in /usr/share/ash/examples.

The binaries of the examples are installed in /usr/bin/ASH*

If you need to reinstall the API, start by removing the previous one using:

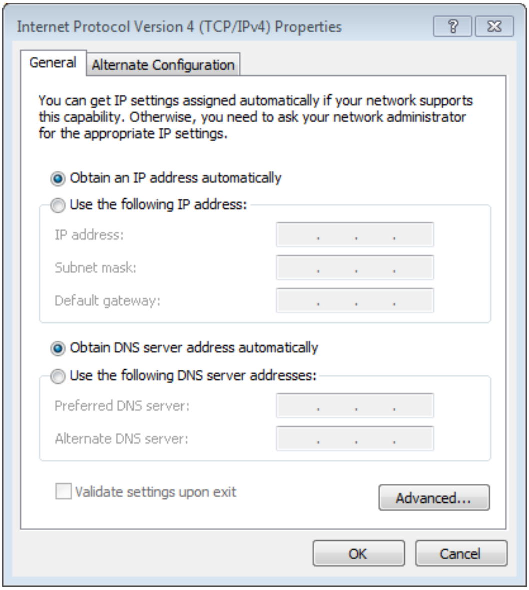
```
$ sudo dpkg -r ash
```

8 Network Configuration

The following configuration description is for the factory configuration (Avahi)

8.1 Windows

Configure the Ethernet Network Adapter used to communicate with the head to obtain an IP address automatically (ControlPanel\Network & Internet\Network Adapters):



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

● Obtain DNS server address automatically

○ Use the following DNS server addresses:

Preferred DNS server: . . .

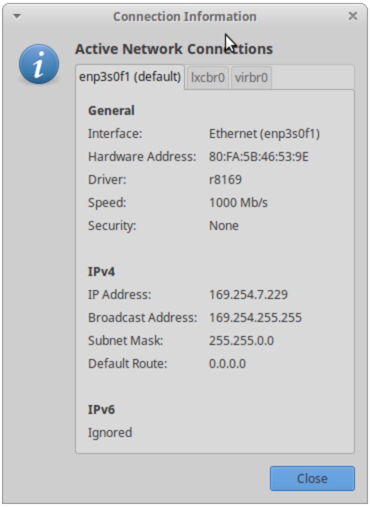
Alternate DNS server: . . .

□ Validate settings upon exit

Figure 7: Ethernet Network Adapter properties on Windows

8.2 Linux

In order to be able to communicate with the head, one needs to configure the CU on the same network (169.254.xxx.xxx). A typical configuration is presented below (System Settings/Network/Wired Connections):



General
Interface: Ethernet (enp3s0f1)
Hardware Address: 80:FA:5B:46:53:9E
Driver: r8169
Speed: 1000 Mb/s
Security: None

IPv4
IP Address: 169.254.7.229
Broadcast Address: 169.254.255.255
Subnet Mask: 255.255.0.0
Default Route: 0.0.0.0

IPv6
Ignored

Figure 8: Ethernet Network Adapter properties on Linux

The network can be configured manually or by running the following commands (avahi):

```
$ ip a
```

Search your network interface (e.g. eth0, enp3s0f1, eth1…) and use it on the command below:

```
$ sudo avahi-autoipd -D --force-bind --no-chroot enp3s0f1
```

Change **enp3s0f1** by your network interface.

If avahi is already running use

```
$ sudo pkill avahi
```

to kill the process and redo the command above.

Note that at each time one disconnects the Ethernet cable of the head (for example when changing from the switch to a direct connection to the CU), the head needs to be rebooted by turning the power OFF and ON.

Also, when performing changes on the network, depending on the network manager in use, one may need to disconnect the network, kill the avahi process and redo the «force-bind» for avahi so that avahi correctly configures the network on the CU.

9 Running Examples

To run the examples, you need a head running and connected to your local network or connected directly to the CU hosting the API (as discussed on Network setup section)

Note that the code of the all examples is available see Building section for details on how to build the examples and your own code.

9.1 ASHDetect

This sample program is used to detect available heads on the local network. It is the first step before running any other sample program.

It takes no argument.

It will simply find available heads on the network using the IDeviceManager detectDevices() method and print the hostname of each of them. The heads listed are the ones which are currently joinable. This program does not detect head in IP static mode. If no head at all is available, a message is printed saying it.

If a head connected to the network is not detected, reboot the head and/or check the network configuration on the PC or if the setup used is supported on the OS (Network setup section).

```
$ ASHDetect
Detecting sensors...
Detected sensor head : head-H-SY-18-01-002.local
Detected sensor head : head-H-SY-19-05-132.local
Detected sensor head : head-H-SY-17-12-149.local
```

9.2 ASHConfig

The *ASHConfig* sample program can be used for two goals:

- Change the current configuration of a head.
  It takes as first parameter the host name of the head we want to interact with (find it using *ASHDetec*t). The second argument is the parameter to update.

```
$ ASHConfig -h
Hostname not specified.
Getter: ASHConfig [hostname|ip]
Setter: ASHConfig [hostname|ip] {-e [msec]} {-f [1-25]} {-l [0-100]} {-x
[extension,optional_parameters]} {-i
[avahi|dhcp|static_xxx.xxx.xxx.xxx/yy]} {-r [reason/comment]}
        ASHConfig [hostname|ip] {-s
[camera_raw_left_right|camera_raw_right_disp]}
        ASHConfig [hostname|ip] {-b [bitstream]}

        hostname:                Name of sensor given by the detect program to
find the name of your sensor.
        ip:                      Ip adress of the head if you use a static ip
adress.

        Volatile paramaters (resets to default after reboot)
                -e, --max-expo-time:    Maximum exposure time
                -f, --fps:              Framerate
                -l, --grey-level:       Target grey level
                -x, --extension:        Call extension (see documentation)

        Persistent paramaters
                -i, --ip-conf-method:   Ip configuration method

        Persistent paramaters inducing a sensor head reboot
                -b, --set-bitstream:    Set current bitstream
                -s, --stream-mode:      Head Streaming Mode
                -r, --reboot:           Force software reboot
```

- Print the current configuration of one head;
  If you want to print the configuration, use *ASHConfig* with no further arguments

```
$ ASHConfig head-H-SY-17-12-149.local

# System configuration :
------------------------
Sensor name:
Product number:         BXT3-SH-CF1-S1-P5-W03
Serial number:          H-SY-17-12-149
FW version:             OMG-2.0.3 (20200210.120058)
Bitstreams available:   sgm_1024x512 sgm_256x128 sgm_512x256
sgm_768x384
Bitstream current:      sgm_768x384
Extensions:             confidence_threshold lr_check_enable
lr_check_max_disparity median_enable overexposure_level
set_confidence_threshold set_lr_check_enable
set_lr_check_max_disparity set_median_enable set_overexposure_level
set_subpixel_scaling subpixel_scaling
State:                  running
Temperature:             60 °C
Ip configuration mode:  avahi
Streaming Mode:          camera_raw_right_disp
extension/lr_check_enable:          1
extension/lr_check_max_disparity: 2
extension/median_enable:            1
extension/subpixel_scaling:         2
extension/confidence_threshold:    28
extension/overexposure_level:      250

# Image configuration :
-----------------------
Framerate:                 14 frames per second
Image size:                 828x544
Image format:               bayer8-rggb
Disparity map:              768x384
Disparity fmt:              Y8
Right focal length (pixel):   179.056
Cameras baseline (mm) :  100.179
Right optical center X (pixel): 245.935
Right optical center Y (pixel): 116.497
Rectified image width :         768
Rectified image height :        384
Max exposure time :             25152
Grey level percentage :         75
```

9.3 ASHViewer

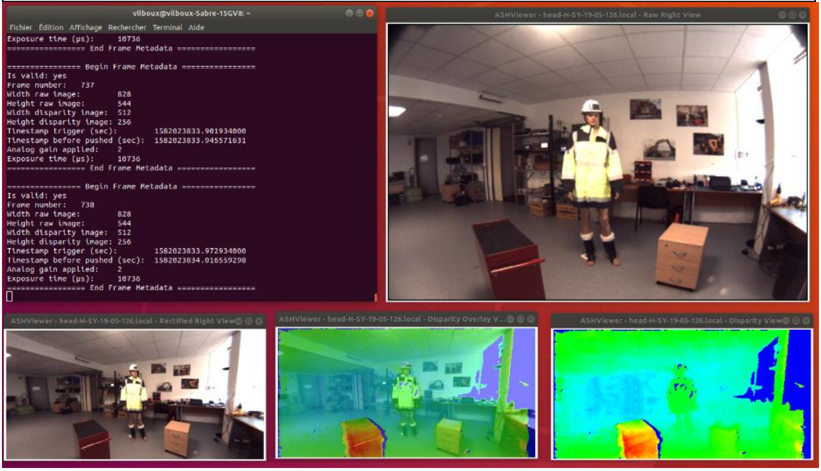
The ASHViewer sample program can be used to images or metadata.

```
$ ASHViewer -h
Arguments : [hostname|ip] [-vdDrlRL]

        The head can not stream both the disparity map and the left image.

        hostname:   Name of sensor. You can use detect program to find the
name of your sensor.
        ip:                IP adress of the head if you use a static ip adress.
        -r --right       Display right raw image
        -l --left          Display left raw image
        -d --disp        Display disparity map colorized. Closer: red, further:
green
        -D --disp_rect  Display same disparity with rectified right image as
background
        -R --rect_right  Display right rectified image using calibration data
        -L --rect_left     Display left rectified image using calibration data

        -v --verbose      Verbose mode: print metadata got for each frames

        -h --help           Display this help

Note that, if no display option is given, raw display is selected by default.
Multiple display options can be used at the same time.
```

Here is a sample of metadata message for one frame received:

```
$ ASHViewer head-H-SY-17-12-149.local -rRdv
streamMode: camera_raw_right_disp

=============== Begin Frame Metadata: ================
is valid : yes
Frame number : 1
Width raw image: 828
Height raw image : 544
Width disparity image : 496
Height disparity image : 250
Timestamp captured (emission) : 102,183828000
Timestamp pushed (emission) : 1576855385,213754739
Analog gain applied : 3
Exposure time (µs) : 11129
```

```
================== End Frame Metadata ==================
```


vilboux@vilboux-Sabre-15GV8: ~
Fichier Édition Affichage Rechercher Terminal Aide
ASHViewer - head-H-SY-19-05-126.local - Raw Right View
ASHViewer - head-H-SY-19-05-126.local - Rectified Right View
ASHViewer - head-H-SY-19-05-126.local - Disparity Overlay V...
ASHViewer - head-H-SY-19-05-126.local - Disparity View

```
Exposure time (µs):     10736
================= End Frame Metadata =================

================ Begin Frame Metadata ================
Is valid: yes
Frame number:   737
Width raw image:        828
Height raw image:       544
Width disparity image:  512
Height disparity image: 256
Timestamp trigger (sec):        1582023833.901934000
Timestamp before pushed (sec):  1582023833.945571631
Analog gain applied:    2
Exposure time (µs):     10736
================= End Frame Metadata =================

================ Begin Frame Metadata ================
Is valid: yes
Frame number:   738
Width raw image:        828
Height raw image:       544
Width disparity image:  512
Height disparity image: 256
Timestamp trigger (sec):        1582023833.972934000
Timestamp before pushed (sec):  1582023834.016559298
Analog gain applied:    2
Exposure time (µs):     10736
================= End Frame Metadata =================
```

Figure 9: ASHViewer frame output

9.4 ASHUpdate

The ASHUpdate sample program can be used to:

- Check if we have an update available for the head requested;
- Actually perform the update.

To run it, you will need a firmware different from the one installed on the head.

It takes as first parameter the host name of the head we want to interact with (find it using ASHDetect).

The second argument must either be “check” or “update”. In the case of a *check*, the program will simply check if the head can be updated with a firmware we have locally in /usr/share/ash/firmware for Linux and on %ASH_FIRMWARE_PATH% for Windows. A firmware file looks like: head_firmware_<version>_<date>.bin.

An update is possible if the current version on the head is not the same as the latest local version. This allows to easily rollback an update.

In the case of an *update*, the program will check if there is an update available, and trigger the update using the latest local firmware. It will then wait for the head to successfully update and reboot. When the head is joinable, the program will print its system configuration to display the change in the firmware version.

```
$ ASHUpdate -h
Arguments : [hostname|ip] { -c } { -u } { -a } { -i [value] }

  hostname:     Name of sensor. You can use detect program to find the
name of your sensor.
  ip:               Ip adress of the head if you use a static ip adress.
  -c | --check:   Check if an update is available
  -u | --update:  Install the last available firmware automatically
  -a | --all:        Display all available firmware files name
  -i  | --install:    Firmware file name to install.
```

```
$ ASHUpdate  head-H-SY-17-12-149.local -u
# System configuration :
------------------------
Sensor name:
Product number:  BXT3-SH-CF1-S1-P5-W03
Serial number:    H-SY-17-12-149
FW version:         OMG-2.0.1 (20191210.132105)
Temperature:      63 °C

Trigger update on head...
The last firmware version is 'OMG-2.0.3_20200210.120058'
Waiting...
..................................................................................................
..................................................................................................
..................................................................................................
...................................................................................



Head is now online.
# System configuration :
------------------------
Sensor name:
Product number:  BXT3-SH-CF1-S1-P5-W03
Serial number:    H-SY-17-12-149
FW version:         OMG-2.0.3 (20200210.120058)
Temperature:      63 °C

Done new firmware version: OMG-2.0.3 (20200210.120058)'  target
was:'OMG-2.0.3_20200210.120058'
```

10 Building examples (Compilation)

10.1 Windows

First of all, set up a development environment by installing these tools:

- CMake version 3.x
- Visual Studio Build Tools 2015 (For Windows 7, it requires Service Pack 1 and .Net 4.5.1)
- Qt 5.7.1 with Pre-build libraries for MSVC 2015 64 bits

Then, locate qtenv2.bat in your Qt installation folder that matches your build tools (here MSVC 2015).

For example, if the Qt installation folder is located in C:\Qt\Qt5.7.1 and pre-build libraries for MSVC 2015 64-bits have been installed, then the full path to qtenv2.bat would be C:\Qt\Qt5.7.1\5.7\msvc2015_64\bin\qtenv2.bat.

Open it with any text editor and append the following line:

```
%comspec% /k ""C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\vcvarsall.bat"" <arch>
```

Where *<arch>* could take amd64 or x86 for Qt 64 or 32-bits respectively.

Finally, test your development environment by opening the Qt Command Prompt.

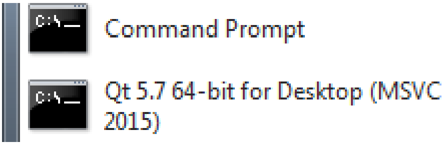


Figure 10: Qt5.7 64-bit command prompt for MSVC 2015

Type the command below, it should return the full path to you MSVC compiler cl

```
$ where cl
```

If everything is in proper place, then you can start building the examples:

```
$ cd “%ASH_PATH%\share\ash\examples\viewer”
$ mkdir build
$ cd build
$ cmake .. -G “NMake Makefiles” -
DCMAKE_PREFIX_PATH=”%ASH_PATH%\share\ash\cmake” -Wno-dev
$ nmake
```

10.2 Linux

Just copy the exemple source code then compile it:

```
$ cp -r /usr/share/ash/examples /tmp/omega_exemples
$ cd /tmp/omega_exemples/viewer
$ mkdir build
$ cd build
$ cmake ..
$ make
```

11 Technical characteristics

11.1 CU specifications

Minimum suggested specifications for the CU x86_64b running the API:

- **Architecture**: x86_64 (64-bits)
- **Processor**: Intel i3, equivalent or greater
- **GPU**: not required but recommended for accelerated UI
- **RAM**: At least 4 GB
- **Network**: Ethernet Gigabit
- **OS**: See Supported OS section
- **Free disk Space**: > 10 GB of free disk space total

Typical resources used when streaming video (using an Intel core i3-4150 CPU 3.5GHz)**:**

- CPU: 25%
- Bandwidth: 300Mbit/s
- Memory: 100MB

API available for some embedded ARM devices

- Nvidia Xavier
- Nvidia Jedson
- Nvidia Nano
- Raspberry Pi 4

11.2 Omega Head specifications

Input voltage	9-36VDC (using connector box)
Power	7W
Connector box	3 connectors IP67: input voltage (with 2 meters cable); head communication and power (with 5 meters cable); Ethernet for communication with processing unit (optional)
Head dimensions	186,8 x 75,95 x 64,6 (mm)
Fixings	4 x Screws with bracket (optional)
Weight	1,2 Kg
Operating temperature	-40 to +75°C

IP	IP69K for the head housing (Totally protected against dust, withstands jet cleaning) & IP67 for the deported connector
Vibration	IEC 60068-2-64: Vibration, broadband random and guidance
Shock	IEC 60068-2-29: Impact test on equipment subjected to repeated shocks
Type of sensor	Color only
Image sensor model	e2v Sapphires color
Sensor technology	CMOS
Sensor resolution	1280x1024 pixels
Sensor format	1/1.8´´
Sensor responsivity	6600 LSB$_{10}$/(lux.s)
Sensor lenses aperture	2
Sensor lenses dist	f-θ type
Sensor dynamics	8 bits
Horizontal field of view	120°
Vertical field of view	90°
Raw images size	828x544 pixels
Rectified image	See disparity (not send, computed by the API)
Disparity size	1024x512 pixels (SGM version) 768x384 pixels (SGM version) 512x256 pixels (SGM version) 496x250 pixels (Census version) 256x128 pixels (SGM version)
Disparity distance range	full range from 30cm to 70m (for 512x256 disparity), typical working zone from 35cm to 10m
Calibration	Sensors are factory calibrated

The following calibration data is available from the API Reference Guide:

- LUT - lookup table used to rectify the raw image
- RightFocalLength - focal length of the right sensor in pixels
- Baseline - distance between both sensors in millimeters
- RightOpticalCenterX - Right optical center X in pixel
- RightOpticalCenterY - Right optical center Y in pixel
- RectifiedImgWidth - width of the rectified image in pixel
- RectifiedImgHeight - height of the rectified image in pixel.

Complete listing of standards applicable to the Omega head[1]:

- IEC 60068-2-64: Vibration resistance
- IEC 60068-2-29: Shock resistance
- IP69K DIN 40050-9: Protection index
- NF-ISO-EN-CEI ISO16750-3: Operating temperature
- NF-ISO-EN-CEI EN60068-2-14: Thermal shock
- ISO 13766 (2006): Earth-moving Machinery - Electromagnetic Compatibility (20-1000 Mhz - AM 80% 1kHz - 30V/m)
- ISO 11452-2 (2004): Road vehicles - Electromagnetic Compatibility
- ISO 14982 (2014): Agricultural and forestry machines - Electromagnetic Compatibility (20-1000 Mhz - AM 80% 1kHz - 30V/m)
- EN 12895: Industrial trucks - Electromagnetic Compatibility (On contact ±8kV In air : ±15kV)
- EN 13309: Construction machinery - Electromagnetic Compatibility (On contact ±6kV In air : ±8kV)
- EN 55022: Information Technology Equipment - Electromagnetic Compatibility
- EN 61000-4-3: Electromagnetic field immunity

[1] Certifications performed in the context of the Blaxtair® product (processing unit and head), not Omega (isolated head), but are also applicable to the head Omega due to hardware equivalence.

12 Troubleshooting

Q: On Linux, I obtained the following error "We did not find the main interface on this computer. Do you have avahi-autoipd running?" while trying to run ASHDetect.
A: Configure your network interface using avahi-autoipd, read Network configuration on Linux.

Q: On windows, I obtained the following error "C:\Program Files (x86)\ash 2.0.X\bin\scripts\tools\dig.exe: parse of C:\Windows\system32\Drivers\etc\resolv.conf failed" while trying to run ASHDetect.
A: Copy the file resolv.bat from %ASH_PATH%\bin\scripts\etc to %WINDIR%\system32\drivers\etc

Q: On windows, when running ASHDetect example, I always obtain "No sensor detected" message even if I had a single head connected to the network.
A: Make sure your network setup is supported by windows. Don't forget to check that you have only one enabled network adapter (disable all physical/virtual adapters) and that the head is connected directly to your PC. After that, turn OFF/ON the head by unplugging/plugging the power supply. Then check Network configuration on Windows.

Q: I have just received a new firmware for the head, how can I perform an update?
A: The new head firmware should be placed in /usr/share/ash/firmware for Linux and %ASH_FIRMWARE_PATH% for Windows. The fastest way to update the head is to run the ASHUpdate example.

13 Frequently Asked Questions

Q: What is the max latency of images (Raw & Disparity)?
A: Latency depends on disparity resolution, indicative values are (world <-> image available on API): < 200ms for 1024x512 disparity and < 140ms for 512x256 disparity.

Q: How many heads can I use in my local network?
A: Can use up to 10 heads. It depends on your network setup because each head needs ~115Mbit/s.

Q: Can I use the heads through internet?
A: No, the heads are not accessible through routers, modems, etc.

Q: Can I update the head when a new firmware is available?
A: Yes, using the API with this command:

```
$ ASHUpdate head-<HEAD-S-N>.local -u
```

Q: What is the max frames per second available on the head?
A: Stable up to 18 fps, it can be less depending on disparity resolution.

Q: How long does it take to boot the head?
A: Starting time is 35 seconds.

Q: What parameters can I set?
A: Main parameters are: mean image grey level, frames per second, maximum exposure time, network configuration, streaming mode and disparity resolution.

Q: What information can I read from the head?
A: The information available for read is: firmware version, video sensor P/N, video sensor S/N, focal length, lookup table to compute rectified image (LUT), distance between both sensors, X coordinate of the optical center, Y coordinate of the optical center, rectified image width, rectified image height, image sizes, CPU temperature, exposure time, sensor analog gain, image acquisition timestamp and timestamp before sending data by Ethernet. Accessible with this command:

```
$ ASHConfig head-<HEAD-S-N>.local
```

Q: How to change the disparity resolution?
A: With an Omega version HD, by using ASHConfig, see all bitstreams available, then select the corresponding resolution:

```
$ ASHConfig head-<HEAD-S-N>.local –b sgm_768x384
```

Q: Are there program examples available?
A: Yes, four example programs are available: Heads detection, display video and disparity, set and read parameters, update head’s firmware.

Q: Are there program examples with code available?
A: Yes, all example programs have their code available.

Q: Do I need administrator rights to use the OMEGA API?
A: Yes.

Q: How is the API installed?
A: By using a debian package for Linux and an executable for Windows.

Q: Can I run the API on a virtual machine?
A: Using virtual machines is not guaranteed to work.

Q: What programming languages can I use?
A: The programming interface is written in C++.

Q: Is the API multi-thread and asynchronous?
A: No, the API is not multi-thread safe and all functions are synchronous.

Q: What is the API Licence?
A: You can use the API for commercial use (see licence for details).

Q: What is the resolution of the raw image?
A: The resolution is 828x544.

Q: What is the size of the disparity image (distance map)?
A: The size is configurable for the HD version: 1024x512, 768x384, 512x256, 256x128 pixels. For the standard version, only the 496x250 resolution is available.

Q: Can I connect the head to the battery of a machine?
A: Yes, using the connector box that stabilizes the power.

Q: What is the power input voltage accepted?
A: 9VDC to 36VDC accepted by the connection box.