PID2021-125051OB-I00 HVD (2022-2025)

Harvesting Visual Data: enabling computer vision in unfavourable data scenarios

D1.1 v1

System Infrastructure

Video Processing and Understanding Lab

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Supported by

AUTHORS LIST

José M. Martínez	josem.martinez@uam.es
Juan Carlos San Miguel Avedillo	Juancarlos.sanmiguel@uam.es

HISTORY

Version	**Date**	**Editor**	**Description**
0.1	16/08/2023	José M. Martínez	First Draft version for contributions
0.2	12/09/2023	Juan Carlos San Miguel	Contributions
0.3	13/09/2023	José M. Martínez	Working Draft edition
0.4	14/09/2023	Juan Carlos San Miguel	Working Draft comments
0.5	14/09/2023	Juan Carlos San Miguel	Final Working Draft
1.0	18/09/2023	José M. Martínez	Editorial checking

CONTENTS:

1. Introduction

This document describes the System Infrastructure of the HVD project that will be used for the design, developments and deployment of use cases during the project within WP2 and WP3.

This current first version will be updated, if required, yearly during the project lifetime.

It contains the following chapters:

- Chapter 1: Introduction. A short introduction to the document.
- Chapter 2: Previous System Infrastructure. Describes the System Infrastructure available at the VPULab before the HVD project.
- Chapter 3: Acquisitions. Describes the acquisitions done during the project so far.
- Chapter 4: Current System Infrastructure. Describes the System Infrastructure available for the project after new acquisitions and, if applicable, removal of some existing ones.
- Chapter 5: Conclusions. It describes the capabilities and shortcoming of the current System Infrastructure and potential additional needs.

2. Previous System Infrastructure

2.1. Infrastructure for image capture

This section describes the devices available for capturing video sequences, both in terms of cameras of various types as well as scenarios and lighting systems.

2.1.1. Fixed cameras

The current fixed camera infrastructure dates to 2005, when a system of three machine vision cameras was installed in two access corridors of the Polytechnic School (see Fig. 1). One of the accesses has two cameras, to consider stereo-vision applications; the other has a single camera with variable focal length, which allows varying the region of interest. Both consider scenarios with natural, not artificial, illumination, scenarios especially suitable for people counting, since their usual use is as corridors connecting two buildings of the High Polytechnical School.

These are digital cameras, model DFW-X710 from SONY, two of them with fixed optics and the third with variable optics. The cameras send uncompressed image sequences (1024x768, color, 15 fps.) through an IEEE1394 bus, extended by a fiber optic link (GOF), to a rack of PCs. Each PC receives the signal from up to two cameras. The rack interconnects the receiving PCs with a disk server via a Gigabit Ethernet network, connected to the Internet and protected by a firewall.

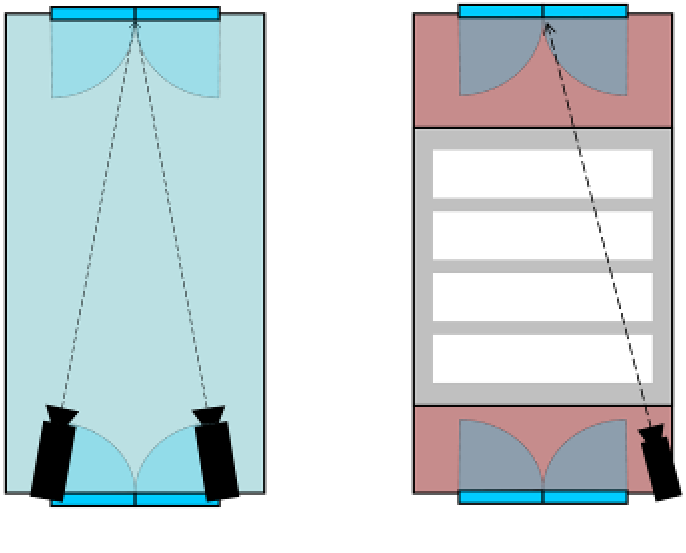


Figure 1 Placement of fixed cameras on floor 3 (left) and first floor (right) in the EPS.

Figure 2 Real setup of fixed cameras on floor 3 (left) and first floor (right) in the EPS.

2.1.2. PTZ (Pan-Tilt-Zoom) cameras

The current infrastructure dates back to 2006, when a system of three PTZ cameras was installed in the lobby of the Polytechnic School (see diagram in Fig. 3, aspect of the covered area in Fig. 4, and detail of a camera in Fig. 5). The scenario considered by this installation is multiple. On the one hand, a scenario in which a large, non-transparent area is recorded (due to the four central columns) with a high density of people circulating through it and with partially natural lighting, since the front of the hall is glazed; on the other hand, taking into account that the side cameras can be oriented towards the opposite end of the hall, which has two access corridors to the classrooms, a suitable scenario is considered for counting in situations of medium density and with mainly artificial lighting. In both cases the high reflectivity of the floor poses additional complications to the analysis.

These are digital IP PTZ cameras, model SNC-RZ50P from SONY. They have several operating modes and their position and focal length can be controlled by software. The cameras send compressed image sequences (JPEG) or directly a video stream (MPEG4 or H264) through an Ethernet network, to a Gigabit Ethernet router, connected to the Internet and protected by a Firewall.

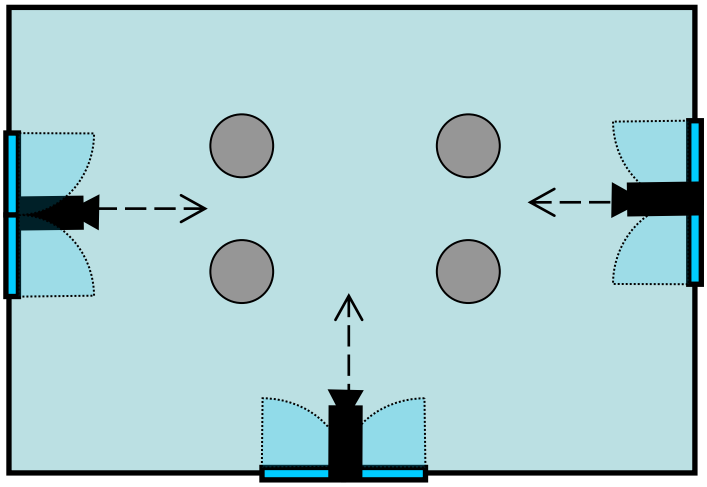


Figure 3 Diagram of PTZ cameras placement in EPS hall.

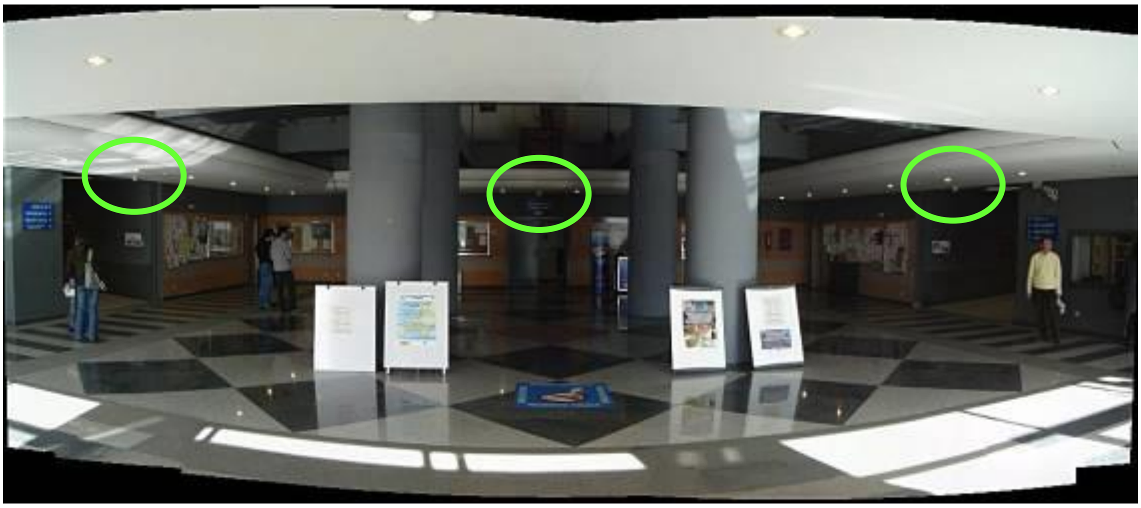


Figure 4 Diagram of the PTZ cameras (left) and detail of one of the mounted cameras (right).

Figure 5 Detail of one of the PTZ cameras located on the ceiling of the EPS hall.

Figure 6 Detail of the PTZ camera for various uses and installation on a mobile platform.

2.1.3. Hand-held vision cameras

The current portable vision camera infrastructure dates back to 2007. It consists of two pairs of high resolution (1600x1200 and 1900x1080) and quality (uncompressed capture) cameras, as well as sets of lenses and brackets to configure a portable multi-camera capture system. The scheme is particularly suitable for reduced scenarios, such as intelligent rooms, stereo vision systems, etc.

These are digital cameras, models Pioneer piA 1600-35gm/gc and piA 1900-32gm/gc from BASLER (see detail in Fig. 7). They feature various modes of operation in terms of resolution and frame rate.

The cameras send uncompressed image streams over a Gigabit Ethernet network, usually directly to a laptop.

Figure 7 Detail of one of BASLER's high-resolution vision cameras.

2.1.4. Depth vision cameras

Apart from the devices already described, the Group has special capture devices that can help solve vision problems in complex situations or contexts.

One of them is the Kinect device for Xbox 360 from Microsoft © (see details in Fig. 8). From the vision point of view, the device allows to obtain RGB images of a scene, including for each pixel the information of the depth at which it is located from the capturing devices.

Figure 8 Detail of a Kinect camera and its placement.

2.1.5. Infrared laser structured light capture system

The Group has a system for the emission of non-visible structured light (based on the use of an infrared laser and diffraction gratings), and a camera to capture this light (conventional camera without infrared filter and with a band-pass filter centered on the laser band).

2.1.5.1. Infrared structured light emission

The light source is an infrared laser. It is the Z-Laser model ZM-18H (see Figure 9). It works at a wavelength of 808 nanometers, corresponding to the infrared frequency range, which is invisible to the human eye, thus preventing it from interfering with the scene on which it acts. Its nominal power of 100 mW is high, which makes it necessary to take into account the safety values of the regulations for laser devices. A device has been developed to regulate the power emitted by the laser.

Figure 9 Detail of infrared Z-Laser.

To obtain a structured light pattern, the laser is fitted with a lens that ideally has the passband at the same wavelength (to avoid losses). Fig. 10 shows examples of available patterns.

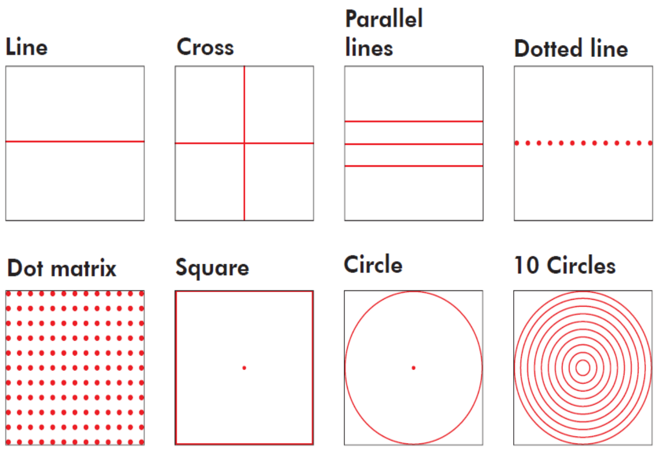


Figure 10 Structured light patterns generated by laser lenses.

2.1.5.2. Infrared structured light capture

The basic capture element is an IP camera, model CDN-62 with SONY Exview HAD 1/3" CCD, chosen because it is particularly sensitive to infrared light. This camera has a mechanical IR filter, which is deactivated when placed in night mode (see Fig. 11).

Figure 11 CDN-62 camera (left) and infrared filters (right).

In order to obtain maximum sensitivity to the structured light pattern, an infrared bandpass filter as centered as possible in the laser operating band is placed in front of the camera lens. Two filters are available for this purpose:

- FB810-10: Filter centered at 810 nm with a passband of 10 nm, i.e. it works between wavelengths of 800 and 820 nanometers.
- FB800-40: Filter centered at 810nm with a passband of 40 nm, i.e. it works between wavelengths of 760 and 840 nanometers.

2.1.6. Handheld battery-powered vision cameras

The group also has devices that emulate video security cameras equipped with WiFi connectivity and powered by lithium batteries. Additionally, they can perform off-line recording, for those cases in which unattended video capture is needed for several hours. It is considered that the current cell phones meet these needs (capture, processing and embedded communication; battery;...) and four cell phones model Xiaomi Redmi Note 4 3GB/32GB are available, with the following characteristics:

- Android 6.0 (Marshmallow) MIUI V8 operating system.
- Qualcomm Snapdragon 625 Octa-core processor
- RAM/Internal memory: 3GB/32GB
- Capture resolution: Image (max 13MP), video (720p)
- Connectivity Wifi 802.11ac and Bluetooth 4.2
- Battery 4100mAh

Figures 12 and 13 show examples of the acquired device and its assembly to emulate a portable video surveillance system. For the development of this infrastructure, previously available tripods were used, and adapters were purchased for mounting.

Figure 12 Redmi Note 4 3GB/32GB device available.

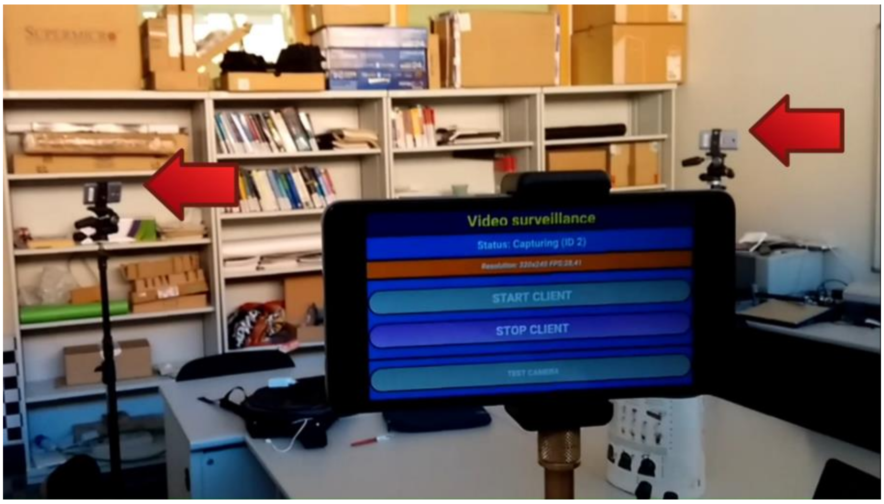


Figure 13 Video surveillance system using tripods and adapters with mobile cameras.

2.2. Infrastructure for distributed image processing

The research group has developed an architecture that supports the development and execution of video sequence analysis applications. The term by which this development platform is called SMCS (Smartphone MultiCamera System).

SMCS establishes a distributed environment for the simultaneous intercommunication of multiple video sources (i.e. smartphones) with processing algorithms running on a server, and the formalized inclusion of contextual information in the analysis process, enabling data streams to be processed in real time.

2.2.1. SMCS Modules

In the first phase, the software developed considers the use of cell phones as sensing units (i.e., cameras). The video surveillance system is composed of three modules (see following figure): cameras, server and client application.

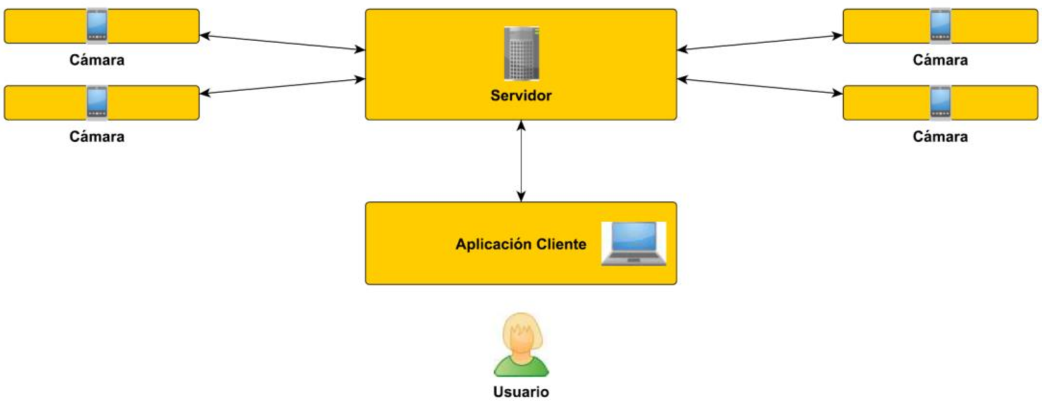


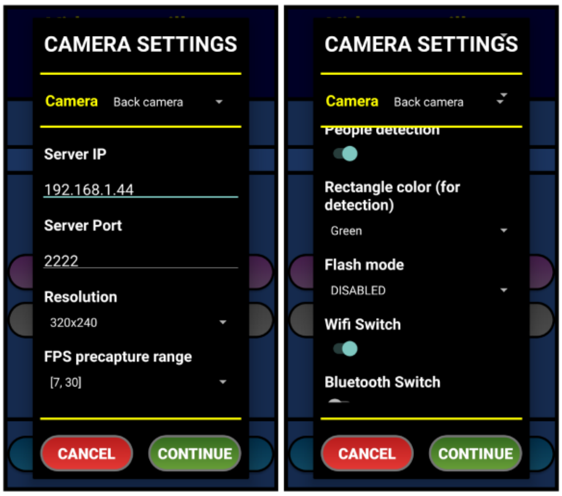
Figure 14 Conceptual architecture of the SMCS platform.

The cameras will be the Android mobiles in charge of connecting to the server and listening to requests coming from the server to perform actions such as configuring recording parameters, controlling wifi or bluetooth connections, sending information about the current status, starting to capture images and sending these images to the server. The server is the intermediate module between the client application and the cameras. The client application is in charge of sending commands to the server, which will be in charge of adapting these commands to requests to the different cameras, obtaining their responses and sending this information to the client application. The client application is in charge of sending the commands to the cameras through the server. The fact of having an intermediate server makes it much easier to implement communication with multiple cameras simultaneously.

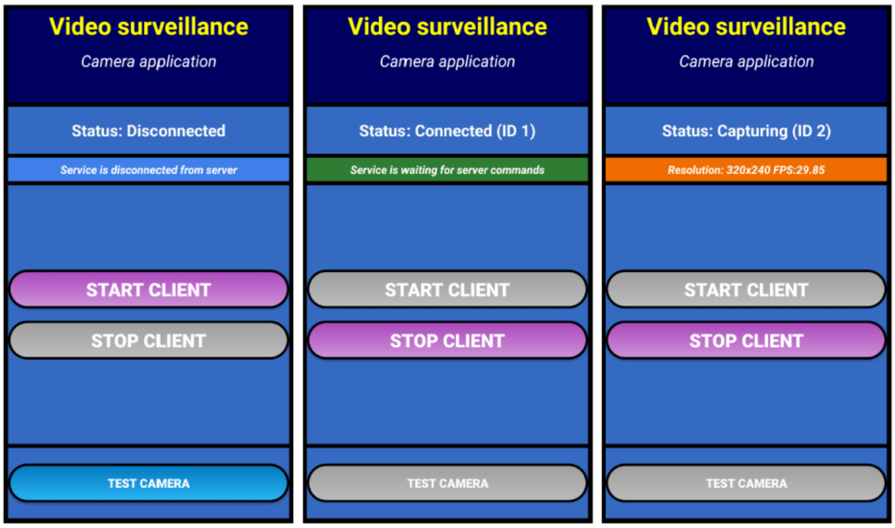
Android Camera

In the development of the mobile application for capturing, processing and sending images, the Android Studio development environment has been used for the Java implementation of the app with the OpenCV image processing libraries included in the executable generated as the final product.

It has a graphical user interface that informs at all times of the status of the mobile device, in particular the status of the camera and the connection to the remote server. In addition to visual functionality, it also has a complex organization that offers from an API with many functionalities related to the camera, connection and image processing to an intelligent management of the phone's resources to respond quickly to commands received from a user, through the server, creating two threads for parallel execution: one that manages everything related to the connection with the server and another that manages everything related to capturing, converting and sending images. Figure 16 shows an example of the developed application.



(a)



(b)

Figure 15 Application developed to control Android cameras/mobiles remotely.

Server

The server (see following figure) is the heart of the system. The server is in charge of managing multiple cameras, interacting with the client application and checking the integrity of each command to gain reliability and robustness. It is the module responsible for the stability of the whole system, since all communications have to go through it. If the server were to go down, the whole system would go down.

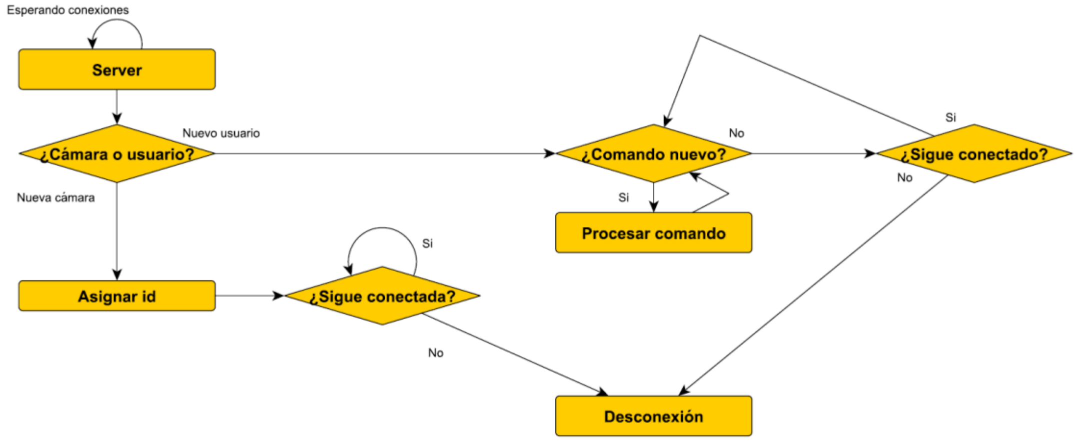


Figure 16 Server' flow diagram.

Applications

An API has been developed to implement applications in C/C++ using OpenCV, whose functionality is listed in the following Table.

	Parámetros	Utilidad
Connect	Cam/User	Conecta una cámara o un usuario al servidor, que le asignará un identificador único
List	Cam/Cmd	El servidor devuelve la lista de cámaras conectadas o la lista de todos los comandos disponibles
Get-current	ID param	Obtiene el valor de un parámetro de una cámara. Puede ser utilizado para devolver todos los parámetros de una cámara o de todas también
Set-current	ID param value	Configura un parámetro concreto para una de las cámaras o todas las conectadas
Start	ID	Comienza a capturar imágenes en una cámara o en todas las que estén inactivas
Stop	ID	Detiene la captura de imágenes en una cámara o en todas las que estén capturando
Get-frame	ID	Obtiene la última imagen disponible en la cámara especificada o en todas
Exit		Termina la conexión con el servidor
Quit		Este comando solo se utiliza en ocasiones que deseamos hacer pruebas y no se accede desde ninguna aplicación. Desasigna el identificador asociado a la conexión pero no cierra la conexión con el servidor. Puede permitir a un usuario dejar de ser usuario y pasar a ser cámara.

Table 1 – API description

2.2.2. SMCS communication

The following is a brief description of the design of the communications between the system modules. These communications can be visualized in the following Figure.

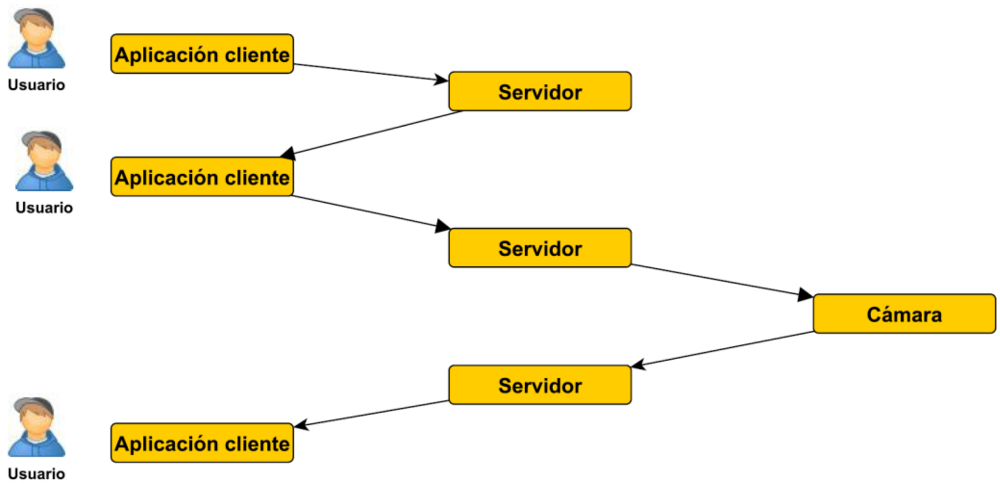


Figure 17 – Communication Flow in SMCS

Communication between Android cameras and server

Android cell phones, i.e. cameras, make an initial connection to the server where they are assigned a unique identifier. They will then remain idle waiting to receive commands. The commands exchanged between the cameras and the server are as follows:

- *Connect*: Connects the camera to the server, which assigns it a unique identifier.
- *Get current*: Gets information on capture parameters, processor and wireless connection status.
- *Set current*: Defines a new value for one of the configurable parameters mentioned above.
- *Start*: Starts capturing images using the previously configured values.
- *Stop*: Stops the image capture started by the previous command.
- *Get Frame*: Once the image capture has been started with the start command, this action can be performed to obtain the last available image.

Communication between server and clients

The client application is in charge of sending the actions requested by the user to the cameras through the server. In this way, the commands of the previous section can be sent to the server, specifying the identifier of the specific camera to be sent.

2.2.3. SMCS API for Android

We have different Android functionalities that allow us to perform complex actions in a few lines of code. As these functionalities are widely used in the application, it has been decided to make an API to access these functionalities in an efficient and robust way. The functionalities included in this API are:

- Enable/Disable Wifi connection. This option allows us to activate or deactivate the wifi remotely.
- Activate/Deactivate the Bluetooth connection. This option allows us to activate or deactivate the Bluetooth remotely.
- Configure the capture parameters. This method allows us to configure different image capture options such as resolution, FPS limited by software (since in Android there is no such possibility by hardware), autofocus, flash, etc.
- Image conversion in YUV to RGB format and RGB to JPEG compression.

3. Acquisitions

As planned in the project proposal, we acquired the outdated GPU models with newer high-memory professional GPU models (e.g., NVIDIA A40, A100 48-80GB) to meet current processing demands operating in 24/7 mode and a high-end server to handle these new acquired GPU units.

In particular, we acquired the SUPERMICRO SUPERSERVER SYS-420GP-TNR (https://www.supermicro.com/en/products/system/gpu/4u/sys-420gp-tnr), which is an updated version of the model 4029GP-TRT2 already available in the previous infrastructure. The specifications of the server are as follows:

- CPU INTEL ICE LAKE 4314 Dual 16-Core @ 24.GHz
- 512GB RAM DDR-3200
- 2-Port Ethernet 10Gbit
- Up to 10 PCIe GPUs

The following figure shows some images of the acquired server

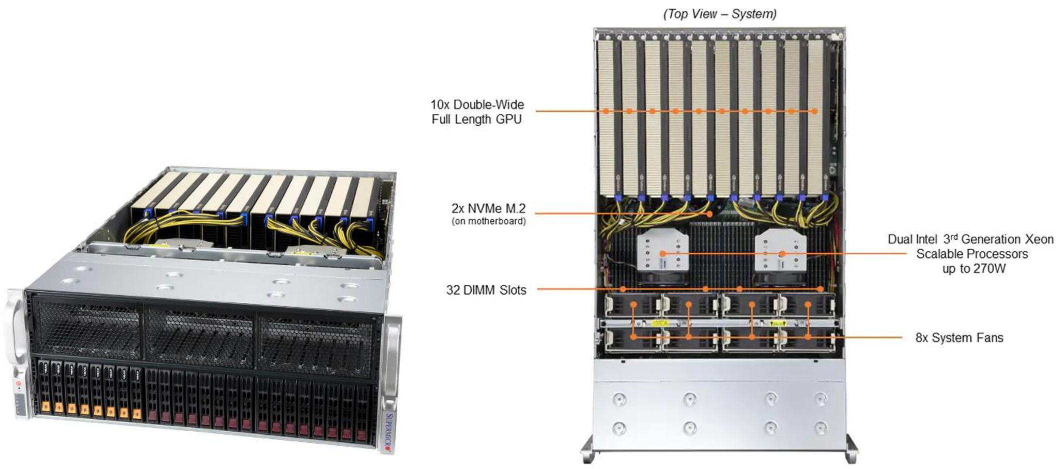
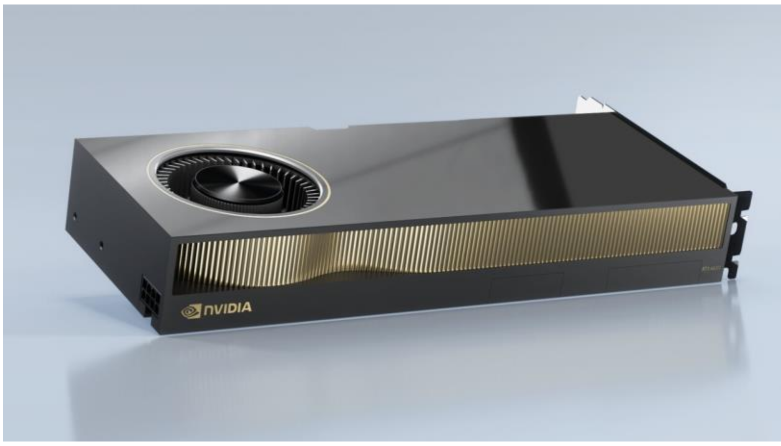


Figure 18 Angled- and top-views of the acquired server SYS-420GP-TNR (Images are taken from https://www.supermicro.com/en/products/system/gpu/4u/sys-420gp-tnr)

Moreover, we also acquired two models of the professional GPUs A40, equipped with 48GB of RAM. The following figure shows the appearance and some details of these GPUs.



Memoria de la GPU	48 GB GDDR6 con código de corrección de errores (ECC)
Ancho de banda de memoria de la GPU	798 GB/s
Interconexión	NVIDIA NVLink 112,5 GB/s (bidireccional) PCIe Gen4: 64GB/s
NVLink	Perfil bajo de 2 vías (2 ranuras);
Puertos Display Port	3 puertos DisplayPort 1.4**
Consumo máximo	300 W
Formato	4.4" (H) x 10.5" (L) Ranura doble
Térmica	Pasiva
Soporte técnico del software de vGPU	NVIDIA Virtual PC,NVIDIA Virtual Applications, Estación de trabajo virtual NVIDIA RTX, NVIDIA Virtual Compute Server, NVIDIA AI Enterprise

Figure 19 Details of the GPU acquired (Images are taken from nvidia.com)

4. Current System Infrastructure

The current system infrastructure is mostly focused on high performance computing oriented exclusively to the storage and processing of massive visual dataset (images and video). This section describes the available equipment for distributing and processing the sequences coming from the capture systems.

4.1. Hardware equipment

The hardware infrastructure is composed of eight rack-mounted servers: two low-end servers for management, two servers with high storage capabilities and four servers with high processing capabilities. These servers are mounted in racks located in a refrigerated room located at the EPS – UAM (see Fig. 14). The list of servers is:

Management (2):

- DELL R210: Dual Core @ 3GHz, 12 GB RAM
- DELL R300: Dual Core @ 3GHz, 24 GB RAM

Storage (2):

- HP PROLIANT G380: Dual 8-Core @ 2.4Ghz, 32 GB RAM
- SUPERMICRO 6049P-E1CR45L: Dual 20-Core @ 2.4Ghz, 92 GB RAM

Processing (4):

- 2x SUPERMICRO SYS-7048A-T: Dual 16-Core @ 2.4Ghz, 160 GB RAM
- 2x SUPERMICRO 4029GP-TRT2: Dual 20-Core @ 2.4Ghz, 384 GB RAM

Moreover, the computing infrastructure has several Graphical Processing Units (GPUs), which are key for training and developing Deep Learning algorithms. The following models are available:

- Nine outdated Graphics-Processing-Units (GPU) models (3x GTX Titan 11GB, 4x GTX 1080Ti 11GB, 2x GTX 2080Ti 11Gb),
- Five recent GPU models (4x RTX Titan 24GB, 1x RTX 3090Ti 24GB).

Figure 20 Detail of processing servers (left) and management/storage servers (right).

4.2. Network architecture

Running on the previously described hardware, the research group has developed a complete server ecosystem for running multiple concurrent virtual machines based on the open-source server solution Proxmox (https://www.proxmox.com). This ecosystem is organized in subnets allowing the following features:

- There is a management subnetwork, oriented to the internal communication of the cluster of physical servers where all the services are hosted. It has an exclusive switch to always ensure the good state of the cluster of the three servers. This subnet is assigned the number 20.
- All physical servers, regardless of their functionality, are placed in a subnet of processing servers, which is identified as subnet 21.
- A service subnet hosts all the services required by the laboratory users. It is located in subnet 22.
- The virtual machine subnet hosts the virtual machines and containers used by the lab's end users, which are located in subnet 23.
- Users connecting through the VPN will have an IP assigned in subnet 24.

The following figure shows a schematic of the network:

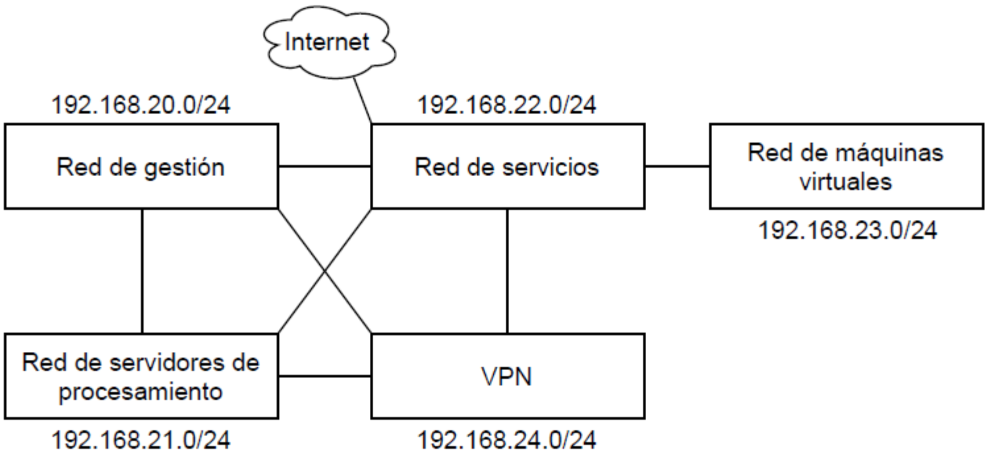


Figure 21. Basic diagram of the main subnetworks and their interconnections

4.2.1. Management network

To host all the necessary network services, two servers are used for management services. These two servers are organized in a cluster to be able to transfer running services transparently. For this purpose, a switch exclusively oriented to the traffic generated by this cluster is used, as shown in the following figure:

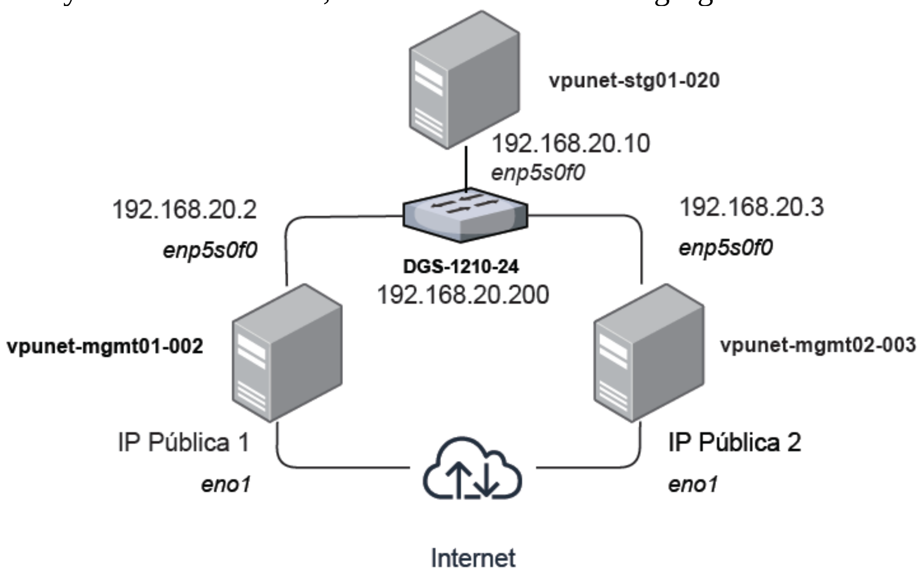


Figure 22. Diagram of the components of the management subnetwork and the different servers that make up the vpunet-mgmt management cluster.

The *vpunet-mgmt01-002* and *vpunet-mgmt02-003* servers are the only ones with Internet connection, therefore, different services are implemented so that certain functionalities in the other servers can work normally.

4.2.2. Host processing network

n addition to the subnetwork for the management cluster, which is called *vpunet-mgmt*, these two previously mentioned servers are in a subnetwork of processing servers, together with other servers destined to work machines, which are part of another cluster called *vpunet-work*. To guarantee a certain scalability and order in the future, a distinction is made within the subnet for each type of server and its role in the network.

- Service-oriented servers have IP addresses between 192.168.22.2 and 192.168.22.9.
- Servers for data storage and network file sharing have IP addresses between 192.168.21.10 and 192.168.21.19.
- Servers for running virtual machines for users are located at IP 192.168.21.20 and above.

The following figure shows the current configuration of the data network.

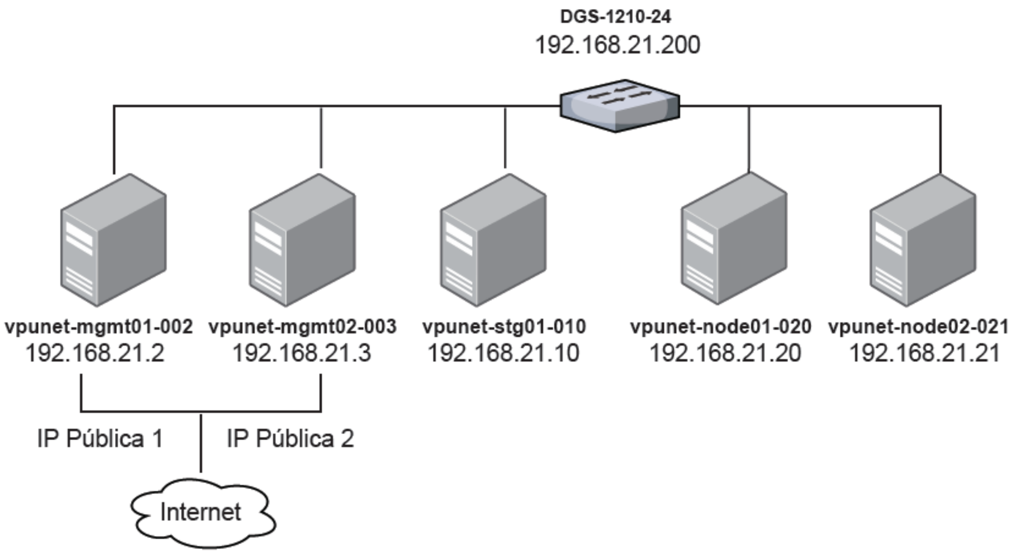


Figure 23 Diagram of the components of the data/server subnetwork (hosts).

4.2.3. Services network

The service network is composed of different task-specific containers. These containers are hosted on one of the three servers located in the management subnet. Each service has an IP address on subnet 22, related to the container identifier on the corresponding Proxmox server.

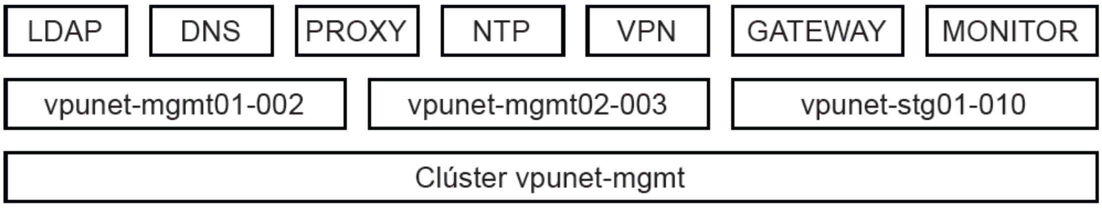


Figure 24 Diagram of the components and containers that are located in the service network. The cluster is comprised of three servers that in turn host some of the services shown as containers).

For this network, labeled as subnet 22, the following services are defined:

- A gateway that allows interconnection between the different subnets previously mentioned. This container requires as many interfaces as subnets are required.
- A name resolution service (DNS) that allows resolving the IPs of the laboratory with the names of the equipment.a proxy that enables Internet traffic to equipment or services that do not have direct access to it.
- An LDAP service to control access to certain virtual machine services. In addition to this, additional services are implemented to allow users and administrators to manage their credentials and accesses in a more direct way than the default implementation.
- An NTP service that allows all computers to keep the same time in a coordinated manner.
- A VPN that allows remote access to the network in a secure and individualized way.
- A monitoring service that records the data of the network resources and is able to send immediate alerts in case of changes in the different systems.

4.2.4. Virtual-Machine network

In the virtual machine network are located all those virtual machines and containers intended to be used for work in the laboratory. These machines are part of a processing-oriented server.

The following figure shows how the virtual machines are organized within the network and how the subnets interact so that the machines get access to the Internet through a proxy.

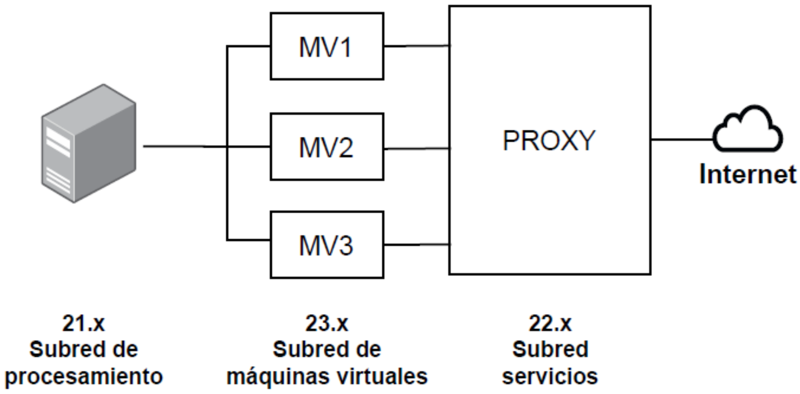


Figure 25 Example of interaction between several subnets for internet access of a virtual machine via a proxy.

5. Conclusions

This document presents the existing infrastructure of the VPULAB research group at different levels: image capture, distributed processing, existing software and high-performance computing. These resources are in exclusive use for the researchers of the group, which includes all the members of the HVD project. This infrastructure is used simultaneously by permanent academics, non-permanent staff and students associated with the group (approx. 25-30 users per year). The updated infrastructure has nine rack servers (approx. 300 cores), 16 GPU (Graphics Processing Units), and a total storage of 200 TB. VPULAB also has 4 GPU-capable PCs.

As future work in the second half of the project, we also consider improving the server ecosystem by creating a processing cluster to maximize the usage of computing resources (employing existing servers and the acquired one) and also by deploying a cluster management software (e.g., SLURM). This cluster is considered a key element for underpinning the development of complex algorithmic solutions, making use of multiple GPUs and also for sharing these GPU resources with the researchers of the project. Moreover, we also consider improving the storage capabilities of the project to meet the increasing demands of recent deep learning developments.