

Document downloaded from:

<http://hdl.handle.net/10251/185958>

This paper must be cited as:

Marfil, D.; Boronat, F.; González-Salinas, J.; Vidal Meló, A. (2021). Design and assessment of a scalable and customizable low-cost tiled display system. *IEEE Latin America Transactions*. 19(5):708-716. <https://doi.org/10.1109/TLA.2021.9448284>



The final publication is available at

<https://doi.org/10.1109/TLA.2021.9448284>

Copyright Institute of Electrical and Electronics Engineers

Additional Information

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Design and assessment of a scalable and customizable low-cost tiled display system

D. Marfil, F. Boronat, *Senior, IEEE*, J. Gonzalez, A. Vidal

Abstract—In this paper, a scalable hardware and software architecture for low-cost tiled display systems (a.k.a. videowalls), customizable through a dynamic web-based management and configuration service is presented and evaluated both objectively and subjectively. It has been designed for both stored and live broadcast/broadband content, in mosaic or warp distributions. The videowalls can be dynamically configured via web in different ways: the displays can create a single screen of a larger size, or they can be configured in a customized way in order to playout different media contents in different display combinations. As display renderers, low-cost devices are proposed as the main hardware element to obtain affordable systems. As a proof of concept, two prototypes have been implemented, including an accurate synchronization mechanism. Both objective and subjective evaluations have been conducted regarding synchronization accuracy and usability of the configuration service, obtaining satisfactory results.

Index Terms—AMP, media synchronization, multi-screen, playout adjustment, tiled displays, videowall.

I. INTRODUCTION

LOS sistemas *tiled display* o videowalls consisten en la combinación de múltiples displays, capaces de reproducir contenido de manera sincronizada, actuando así como una única pantalla de mayor tamaño. En un estudio exhaustivo sobre tecnologías para crear pantallas de grandes dimensiones [1], se afirma que los videowalls pueden resultar más económicos que otros sistemas, como los basados en proyectores de vídeo, ocupan menos espacio físico y, además, la corrección del color o la alineación de displays resulta más fácil que cuando se usan varios proyectores de vídeo. Los videowalls se pueden utilizar como paneles comerciales, informativos, de mando y control, de visualización científica [2]–[4] o para escenarios de colaboración y/o teleinmersión [5], entre otros fines.

Los displays que conforman un videowall se pueden distribuir de diferentes maneras (Fig. 1). Por un lado, en una distribución *uniforme* (en mosaico) los displays conforman una matriz de $M \times N$ displays (M filas, N columnas) del mismo tamaño. Por otro lado, en una distribución *disforme* los displays pueden tener tamaños y orientaciones diferentes. Además, se pueden elegir diferentes formas de combinar los displays de un videowall, bien conformando todos ellos una única pantalla de grandes dimensiones (cada display reproduce individualmente una región del contenido); o bien creando diferentes regiones

del videowall (denominadas *secciones*), en las que se reproducen contenidos diferentes e involucrando en cada región a un número reducido de displays del videowall. En la Fig. 1 se muestran varios ejemplos.



Fig. 1. Ejemplos de videowall uniforme (con 1 y 2 secciones) y disforme.

En un videowall, cada display debe reproducir la porción de vídeo correspondiente de manera sincronizada con el resto de displays de su misma sección. Asincronías de pocos fotogramas ya pueden resultar molestos, sobre todo entre displays contiguos o durante cambios de escena o movimientos rápidos.

Aunque existe una amplia variedad de sistemas comerciales de videowalls (Userful, MagicInfo de Samsung, etc.), su coste resulta muy elevado, debido principalmente a que requieren hardware (HW) específico (p.ej. controladores) y software (SW) propietario. Aparte del precio, también presentan otras limitaciones como son [6]: gestión local; dificultad de gestión, de personalización y de ampliación (p. ej., agregando más displays) para usuarios finales no expertos; o la imposibilidad de crear secciones. En [6] se destacan algunos de los desafíos asociados a la hora de proporcionar experiencias satisfactorias en estos sistemas, como son la escalabilidad, la facilidad de configuración y la integración perfecta en su entorno.

En este artículo se propone y se valida la arquitectura de un sistema videowall de bajo coste, escalable y configurable a través de la web, que permite distribuciones uniformes y disformes, la definición de diferentes secciones, así como la reproducción de contenido almacenado y en vivo. Se presentan sus módulos tanto hardware (HW) como software (SW). Además, con el fin de validar su correcto funcionamiento, así como valorar su usabilidad y facilidad de configuración, se han creado dos prototipos y se han llevado a cabo dos evaluaciones, una objetiva y otra subjetiva, respectivamente.

Este trabajo es una extensión del artículo presentado en [7]. La arquitectura preliminar y el prototipo allí descritos han sido mejorados y optimizados, añadiendo nuevas funcionalidades, así como también se presenta una evaluación del sistema. Se incluyen mejoras relevantes, como son un mecanismo de sincronización propio más estable y preciso, ajustes de sincronización menos intrusivos basados en técnicas de adaptación suave del proceso de reproducción, conocidas como *Adaptive Media Playout* o AMP [8] (en [7] sólo se utilizaban técnicas agresivas como saltos y pausas en la reproducción), y

Submitted for review: 29/04/2020. This work has been funded by the Generalitat Valenciana, through the R&D Program “Grants for research groups to be consolidated, AICO/2017” under Project AICO/2017/059.

The authors are with Universitat Politècnica de València (UPV), 46730 Gandia (Spain) (e-mails: {damarre; fboronat; juagos4; avidal}@upv.es).

soporte para reproducir contenido broadcast en vivo (sólo almacenado en [7]). Además, por un lado, se presentan evaluaciones objetivas de la sincronización alcanzada en un prototipo con más displays (16 frente a 6 en [7]), y utilizando contenido con resoluciones hasta 8K (HD en [7]). Por otro lado, se presentan resultados subjetivos sobre el nivel de sincronización percibido, así como sobre la usabilidad del servicio de configuración del sistema. Las principales contribuciones de este artículo son:

- Comparación cualitativa entre el sistema videowall propuesto y otras propuestas de trabajos relacionados.
- Arquitectura HW y SW de un sistema videowall de bajo coste, escalable, con contenido almacenado y en vivo, permitiendo distribuciones uniformes y disformes, así como la configuración de diferentes secciones.
- Configuración del sistema a través de servicio web, ofreciendo así gestión (local o remota) multi-plataforma.
- Implementación de 2 prototipos, uno (4x4) capaz de reproducir contenido 8K y otro (3x2) más fácil de transportar para realizar evaluaciones subjetivas.
- Evaluación objetiva y subjetiva con resultados satisfactorios. No se han encontrado artículos previos incluyendo resultados relacionados con la Calidad de Experiencia (QoE) percibida por los usuarios en videowalls.

El artículo consta de la siguiente estructura: en la Sección 2, se presentan trabajos previos relacionados y una comparativa cualitativa con el sistema propuesto. En la Sección 3, se presenta la arquitectura HW y SW del sistema propuesto. En la Sección 4, se presentan los prototipos implementados, además de las evaluaciones objetiva y subjetiva. El artículo finaliza con las conclusiones presentadas en la Sección 5.

II. ESTADO DEL ARTE

En esta sección se incluye una revisión de trabajos anteriores en los que también se proponen sistemas económicos, además de una comparación cualitativa con el sistema propuesto. No se han considerado sistemas comerciales ni tampoco aquellos trabajos en los que se presentan entornos SW o middlewares ([9]–[14]) que permiten implementar sistemas videowall, puesto que dicha finalidad no es la del presente artículo. Sin embargo, los autores sí comprobaron el funcionamiento del trabajo propuesto en [10], SAGE2, por las similitudes que ofrecía comparado con el sistema propuesto. Cuando se implementó en dispositivos con un rendimiento medio/alto, el funcionamiento fue correcto en cuanto a la reproducción de contenido multimedia. En cambio, cuando se implementó en una Raspberry Pi (en adelante, RPi) modelo 3B, dicho funcionamiento no resultó satisfactorio.

A. Arquitecturas disponibles

Actualmente, existen dos opciones para las tecnologías de transmisión y distribución de contenido en sistemas videowall. Una opción se basa en el uso de HW específico, y el contenido se distribuye mediante dispositivos HW (p. ej., *video splitters*) o interconectando displays mediante buses especiales siguiendo interfaces estándar concretos (p. ej., *DisplayPort*). La otra opción (adoptada en este trabajo) contempla la transmisión de

contenidos a través de redes IP o bien el acceso a contenido almacenado. En [7] se puede encontrar más información, así como las ventajas e inconvenientes de ambas opciones.

B. Estrategias de sincronización

Existen dos tipos diferentes de estrategias de sincronización entre dispositivos: centralizadas (*Master/Slave* o M/S y *Synchronization Maestro Scheme* o SMS) y distribuidas (*Distributed Control Scheme* o DCS) [15][16]. El esquema M/S implica que uno de los dispositivos de reproducción (*maestro*) proporcione su propio tiempo de reproducción como referencia al resto de dispositivos (*esclavos*), que realizarán los ajustes de reproducción adecuados para sincronizarse a dicha referencia. En el esquema de SMS, existe un dispositivo independiente (*maestro*) encargado de recibir información de la reproducción de todos los dispositivos, procesarla y devolverles una referencia para que se sincronicen. Por otro lado, un esquema DCS implica que todos los dispositivos de reproducción intercambian sus datos de temporización, los procesan y calculan una referencia a la que sincronizarse. En [15][16] se puede encontrar una explicación ampliada de estos tipos de esquemas de control de sincronización y criterios para seleccionar una referencia. En [16], los autores proporcionan una comparación cualitativa entre ellos considerando factores como la robustez, escalabilidad, sobrecarga de tráfico, flexibilidad, ubicación de nodos de control, interactividad, consistencia, causalidad, equidad, coherencia y seguridad.

Por otro lado, existen dos tipos de ajuste de los procesos de reproducción: agresivos (saltos y pausas o S&P) o suaves (AMP [8]). Los ajustes agresivos consisten en realizar saltos o pausas en el proceso de reproducción, dependiendo de si el dispositivo está por detrás o por delante de la referencia, respectivamente. Los ajustes suaves consisten en un aumento o disminución suave de la tasa de reproducción del dispositivo, lo cual es menos abrupto y, por tanto, menos molesto para los usuarios [8].

C. Trabajos previos relacionados

A continuación, se presentan, muy brevemente, los trabajos relacionados y, por motivos de espacio, se presenta en la Tabla I una comparación cualitativa entre ellos y el sistema propuesto. En [17], se propone un sistema basado en la virtualización de su servidor de control, que adopta el Synchronized Multimedia Integration Language (SMIL) [18] para gestionar los metadatos asociados a la sincronización espacial y temporal del contenido. Sólo se presenta el diseño sin describir el HW necesario ni ofrecer resultados sobre el rendimiento de la sincronización. De manera similar, en [19] se presenta un sistema videowall basado en una máquina virtual ejecutada en un equipo de rendimiento alto (IntelCore i7 y 12GB RAM), encargado de generar y codificar los contenidos a reproducir. No presentan resultados relativos a la sincronización. En [20] se presenta un sistema videowall basado en RPi que sigue un esquema de sincronización M/S. Aunque es gestionable vía web, no ofrece funcionalidades como la reproducción de contenido en vivo, distribución disforme o la configuración de secciones. Se indica que el mecanismo de sincronización para video alcanza una

precisión de 1 fotograma, pero al añadir el flujo de audio, se comporta de manera inestable. Los sistemas presentados en [21]–[23] se basan en el uso de RPi, permiten distribuciones disformes (sin rotación) e implementan mecanismos de sincronización. No especifican el esquema de sincronización adoptado ni ofrecen resultados respecto al rendimiento de la sincronización. En [24] se propone el uso de un método de sincronización SMS ([15][16]) para videowalls que emplea técnicas de *machine learning* para predecir la latencia de reproducción de los dispositivos, que deben ser dispositivos con rendimiento medio/alto, y con un coste mayor que los propuestos en este artículo. Se presentan resultados con un nivel de sincronización de menos de 10ms utilizando dispositivos Intel Core i3, i5 e i7. En [25] se presenta otra solución de sincronización SMS para sistemas UHD, basada en la predicción de la temporización de las señales de sincronización vertical del vídeo en cada dispositivo, consiguiendo diferencias de 1 trama entre dispositivos.

III. SISTEMA PROPUESTO

En esta sección, en primer lugar, se enumeran las funcionalidades del sistema propuesto, y, a continuación, los elementos, tanto SW como HW, de su arquitectura (Fig. 2).

A. Funcionalidades

El sistema dispone de servicio web de gestión, que incluye:

- Gestión jerárquica de usuarios y control de acceso.
- Configuración de distribuciones uniformes y disformes (incluyendo rotación de displays) con posibilidad de añadir o eliminar displays (escalabilidad).
- Definición de diferentes secciones.
- Creación de listas de vídeos (*playlists*).
- Configuración de la programación temporal de la reproducción de cada sección.
- Selección de fuente de contenido almacenado o en directo (canal de TV digital o *streaming* IP).
- Selección de la salida de audio.
- Carga de contenidos almacenados.

La Fig. 3 muestra un ejemplo de ventanas de configuración de un videowall 2x3 y de su gestión (con dos secciones creadas, 2x2 y 2x1). La descripción de aplicación de gestión web se puede encontrar en [7].

B. Arquitectura HW

La arquitectura HW del sistema propuesto se compone, principalmente, de los siguientes elementos:

- VCPU o *Videowall Content Provider Unit*. Dispositivo de configuración y gestión del sistema.
- RD o *Rendering Devices*. Son dispositivos de bajo coste encargados de gestionar y controlar los procesos de reproducción en cada display. Existirá uno por display.
- Red LAN IP privada de alta velocidad, para la interconexión de todos los elementos anteriores.
- Altavoces y circuito de selección de salida de audio.

1) Videowall Content Provider Unit (VCPU)

El VCPU se encarga de alojar el servicio web de configuración y gestión. Proporciona a los RDs la información del contenido a reproducir. Dicho contenido puede estar en el propio VCPU o en algún otro dispositivo adicional (p. ej., servidor multimedia) conectado a la LAN privada. Por otro lado, también podría estar disponible en remoto, en cuyo caso el VCPU debe poder acceder a él (p. ej. mediante conexión a redes públicas), descargarlo y almacenarlo con antelación para que sea accesible y reproducible por los RD a través de la LAN privada. Para permitir la reproducción de contenido broadcast en vivo, el VCPU debe poder recibir el contenido de TV digital mediante un receptor DVB-T/S y retransmitirlo a los RD. Para poder configurar el videowall de manera remota, se deberá proporcionar acceso al VCPU a través de redes públicas.

2) Dispositivo de renderizado (RD)

Cada RD está conectado, por un lado, a un display del videowall a través de, p. ej., un cable HDMI y, por otro lado, a la red LAN privada para obtener información del VCPU.

Al encargarse un RD independiente de la reproducción en cada display, la resolución máxima de la reproducción del contenido en cada sección del videowall vendrá restringida por: i) resolución máxima de cada porción del contenido que pueda ser reproducida en cada display y por cada RD; y ii) el número total de RD que haya involucrados en cada sección.

3) Circuito de selección de salida de audio

Cuando sólo se disponga de una salida de audio del sistema videowall, para la selección del contenido de audio, se propone implementar dos circuitos *amplificador sumador* (uno para cada canal, L y R) controlados por el VCPU [7]. El manejo de varias salidas (una por sección) se deja para trabajo futuro.

TABLA I
COMPARATIVA DE FUNCIONALIDADES EN LOS TRABAJOS RELACIONADOS

| Trabajo comparado | Secciones de vídeo múltiples | Streaming IP | Sincronización | Distribución disforme | Programación de contenido & Playlists | Configuración dinámica | Configuración basada en web | Prototipo o Resultados |
|--------------------------|------------------------------|--------------|----------------|-----------------------|---------------------------------------|------------------------|-----------------------------|------------------------|
| Jiménez et al. [20] | X | ✓ | ✓ | X | X | ✓ | ✓ | ✓ |
| Kwang-Yong et al. [17] | X | ✓ | ✓ | X | X | ✓ | X | X |
| Bundulis et al. [19] | X | ✓ | X | X | X | ✓ | X | ✓ |
| Pi Wall [21] | X | X | ✓ | ✓ | X | X | X | ✓ |
| Alabdulsalam et al. [22] | X | ✓ | ✓ | ✓ | ✓ | X | X | X |
| Mihalopoulos [23] | X | X | ✓ | ✓ | X | X | X | ✓ |
| Shin et al. [24] | X | ✓ | ✓ | ✓ | X | ✓ | X | ✓ |
| J. Kanda et al. [25] | X | X | ✓ | X | X | X | X | ✓ |
| Proposed System | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Un ✓ significa que se incluye dicha funcionalidad; Una X puede significar que no se incluye o que no se proporciona información al respecto.

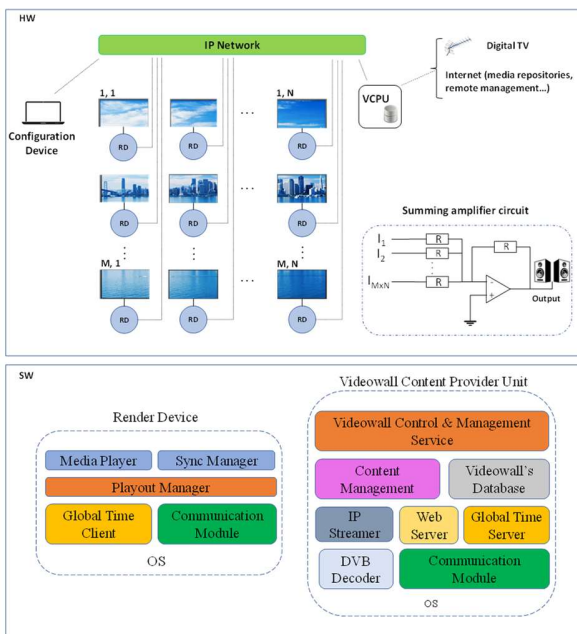


Fig. 2. Elementos HW y SW del sistema propuesto (videowall MxN)



Fig. 3. Ventanas de configuración y de gestión de videowall 2x3 [7]

C. Módulos SW

La Fig. 2 muestra los módulos SW integrados tanto en los RD como en el VCPU. Ambos contienen módulos de comunicaciones y de tiempo global. El RD, además, dispone de módulos de reproducción (*player*) y de gestión y sincronización de la misma. El VCPU, por su parte, dispone de módulos adicionales de gestión de contenido, servidor web, servicio de control y gestión, base de datos, *IP streamer*, y receptor y decodificador de contenido broadcast de TV. La mayoría de dichos módulos, así como el intercambio y el formato de mensajes entre VCPU y los RD, están ya descritos en [7]. Es por ello que en este artículo sólo se describen los módulos añadidos/mejorados.

1) Módulos de recepción y decodificación DVB y Streaming

Por un lado, tal como se ha indicado, para poder recibir y preparar el contenido broadcast de TV digital en vivo, se requiere la instalación de un receptor y decodificador DVB-T/S

en el VCPU. Adicionalmente, en el VCPU se requiere de un retransmisor vía IP (*IP streamer*) para poder transmitir flujos recibidos vía broadcast en directo a través de la LAN privada a los RD (vía *multicast RTP* [27]). Por otro lado, los reproductores de cada RD deben soportar la recepción y decodificación de los flujos multimedia recibidos vía RTP. Respecto a contenidos en vivo, los RD son agnósticos respecto al origen o tecnología original de entrega de los mismos, siempre que el VCPU pueda convertirlo a flujos multicast RTP.

2) Módulo de gestión de la sincronización (*Sync Manager*)

Para conseguir una sincronización precisa, se propone utilizar una referencia de tiempo global (p.ej., NTP [26]) obtenida por el módulo de Tiempo global (Fig. 2, [7]), un esquema de control de los indicados en la Sección 2 (M/S, SMS o DCS), una estrategia para seleccionar una referencia de tiempo con la que sincronizarse ([15][16]) y las técnicas adecuadas de ajuste de los procesos de reproducción. Si se adopta el esquema M/S, uno de los RD de cada sección de videowall se comportará como RD maestro y enviará su propia información de tiempo de reproducción a los otros RD esclavos que ajustarán su proceso de reproducción para lograr la sincronización con el RD maestro. Si se adopta el esquema de SMS, la referencia de sincronización puede ser calculada por el VCPU (*maestro*), a partir de datos que le envíen periódicamente todos los RD, y enviada a todos ellos para que realicen los ajustes apropiados. Si se adopta el esquema DCS, la referencia de sincronización será calculada por cada RD a partir de datos intercambiados entre todos los RD. Independientemente del esquema de control adoptado, los datos e información de control se enviarán por multicast dentro de cada sección.

Cuando un RD obtiene la referencia temporal y detecta una asincronía, deberá corregirla ajustando su proceso de reproducción. Una asincronía se detecta (y corrige) cada vez que el proceso de reproducción de un RD se encuentra por delante o por detrás de la referencia temporal, y sólo cuando dicha diferencia exceda un umbral configurable de asincronía máxima tolerable. En ese caso, el módulo *Sync Manager* del RD forzará un ajuste en el proceso de reproducción para corregirla. Se propone utilizar los dos tipos de ajuste explicados, S&P y AMP, según el nivel de asincronía detectado. Para ello, se definen dos umbrales para comparar el nivel de asincronía calculado, un umbral superior y un umbral inferior (asincronía máxima tolerable). Por un lado, los ajustes mediante S&P se llevan a cabo sólo cuando el valor de asincronía obtenido es mayor que el umbral superior. De esta manera, se trata de corregir lo más rápido posible valores de asincronía considerados excesivos. Cuando el valor esté entre ambos umbrales, se usarán ajustes suaves (AMP).

IV. EVALUACIÓN

En esta sección se describen los prototipos implementados para la validación del sistema propuesto y se presentan las evaluaciones realizadas, una objetiva para comprobar que el nivel de sincronización alcanzado es preciso, y otra subjetiva para comprobar, por un lado, la percepción del nivel de sincronización alcanzado en los prototipos y, por otro lado, que la usabilidad del servicio web de configuración y gestión del

sistema propuesto es satisfactoria.

A. Prototipos implementados

Con el fin de validar el sistema propuesto, se han implementado dos prototipos diferentes, uno para cada tipo de evaluación (Fig. 4). Para la evaluación objetiva se ha implementado un prototipo (prototipo A) con una distribución uniforme 4x4, empleando 16 RPi modelo 3B como RD. Puesto que cada RPi puede reproducir contenido HD, la máxima resolución que es capaz de reproducir este prototipo es de 8K. Es por este motivo que se ha implementado para la evaluación objetiva, ya que, de esta manera, se puede validar el sistema en un escenario donde la exigencia de rendimiento a las RPi es máxima. Debido a que el fin de este prototipo es validar el nivel de sincronización alcanzado, los displays no se han montado en ninguna estructura, y se ha implementado en un laboratorio con 16 monitores HD. Por otro lado, el prototipo implementado para llevar a cabo la evaluación subjetiva (prototipo B), también sigue una distribución uniforme 3x2, utilizando 6 RPi. En este caso, los 6 displays han sido montados en una estructura transportable, para poder situarlo en diferentes puntos del Campus donde trabajan los autores, durante la evaluación.



Fig. 4. Prototipos A (izquierda) y B (derecha) implementados.

1) Elementos HW utilizados

En los prototipos, se han utilizado hasta 16 RPi 3B como RD. Dichos RD se han conectado a 16 monitores Samsung S22D300Hy (22") y a 6 monitores LG M2432 (24") para los prototipos A y B, respectivamente. Como VCPU se ha utilizado un dispositivo Intel NUC (CPU N3050, 4GB RAM y Ubuntu 14.04). Para la recepción de contenido broadcast en vivo de TV, se ha utilizado un receptor DVB-USB WinTV Nova-T en el VCPU. Se han implementado 2 redes IP, la LAN privada y otra adicional para la gestión remota del sistema, mediante un switch D-Link 100Mbps y un router TP-Link 300Mbps inalámbrico conectado a la red del campus, respectivamente.

2) Elementos SW utilizados

Por un lado, para la reproducción del contenido se ha adoptado el reproductor OMXPlayer¹ puesto que está optimizado para la RPi. Además, tal y como se detalla en [7], se adopta el protocolo NTP [26] como proveedor de tiempo global, y se ha instalado en cada RD un cliente NTP y en el VCPU un servidor NTP. Asimismo, en el VCPU también se ha instalado el servidor web Apache para alojar el servicio web de configuración y gestión del sistema. Además, para establecer el canal de comunicación necesario entre los dispositivos involucrados, se ha adoptado la tecnología Websocket [28].

A continuación, se detallan sólo los nuevos módulos SW incorporados respecto a la versión previa del sistema en [7].

a) Módulos de recepción y decodificación DVB y Streaming

En España, el contenido disponible a través de la TV digital (DVB-T) en abierto tiene una resolución máxima de 1080p y

una tasa de 25fps. Por lo tanto, los RD deben soportar necesariamente la reproducción de contenido con dicha resolución. Cabe destacar, que si el contenido que está siendo reproducido en una sección del sistema videowall, no ocupa la totalidad de los displays, se rellena el display automáticamente con márgenes negros donde corresponda.

Para la retransmisión IP de contenido de TV, se ha utilizado el SW DVBlast open-source network streamer², que permite su decodificación y retransmisión mediante *multicast RTP*.

b) Módulo de gestión de la sincronización (Sync Manager)

Para el mecanismo de sincronización implementado en los prototipos se utiliza un esquema de control M/S, basándose en que, según [16], es el esquema que ofrece mejor rendimiento en términos de escalabilidad, carga de tráfico de red e interactividad (retardos de magnitudes muy bajas) cuando se utiliza en escenarios de sincronización inter-dispositivo/destinatario. En cada sección del sistema videowall, existe un RD considerado como *maestro* (M-RD) y el resto de los dispositivos de la sección serán considerados esclavos (S-RD). En la versión previa del sistema, presentado en [7], se adoptó un mecanismo de sincronización de terceros, concretamente uno basado en Python denominado pyOmxSync³. En este artículo, se ha implementado un mecanismo propio más estable y que, además, permite realizar ajustes en los procesos de reproducción de contenidos en vivo. (no contemplados en el mecanismo adoptado en [7]).

A cada sección el videowall se le asigna un grupo multicast con una dirección IP multicast por parte del VCPU en el momento de su creación. El M-RD envía de forma periódica y por multicast a los S-RD de su sección la siguiente información: la posición o instante del contenido que está reproduciendo o información relativa a la ocupación del búfer del proceso de reproducción (para contenido almacenado o en vivo, respectivamente) y el instante de tiempo global (NTP) asociado. La frecuencia de dicho envío es configurable (p. ej., cada 1s). Los S-RD de cada sección reciben periódicamente la información temporal del M-RD necesaria para poder calcular la asincronía respecto al M-RD y decidir si es necesario (si supera el umbral inferior configurado) realizar un ajuste en el proceso de reproducción. En el mecanismo de sincronización implementado, se han adoptado las técnicas S&P y AMP, y se han definido los dos umbrales explicados anteriormente.

Después de que un S-RD haya detectado una asincronía y haya realizado un ajuste en el proceso de reproducción, debe esperar un tiempo determinado configurable (*periodo de guarda*) para poder calcular de nuevo el nivel de sincronización alcanzado y realizar, si fuera necesario, otro ajuste. Durante este periodo de guarda, el S-RD ignorará mensajes con información temporal del M-RD. La Fig. 5 muestra el diagrama de flujo del mecanismo implementado en los dispositivos S-RD.

Respecto a la sincronización de contenidos en vivo, existe una ligera variación en el mecanismo de sincronización. Para este caso concreto, únicamente se adoptan técnicas de ajuste AMP, puesto que realizar ajustes basados en S&P pueden

¹ <https://www.raspberrypi.org/documentation/raspbian/applications/omxplayer.md> (último acceso: abril 2020)

² <https://www.videolan.org/projects/dvblast.html> (último acceso: abril 2020)

³ <https://github.com/markkorput/pyOmxSync> (último acceso: abril 2020)

conducir a situaciones de falta de contenido en el búfer de reproducción. Además, al tratarse de contenido en vivo y que todos los RD de la sección reciben el contenido (*IP stream*) de la misma fuente (VCPU) en la misma LAN, para las posibles situaciones en las que se detecte una asincronía, es muy probable que realizando un ajuste a través de técnicas AMP se corrija satisfactoriamente, sin necesidad de recurrir a técnicas agresivas. En este caso también se tendrá en cuenta el tiempo de guarda después de cada ajuste realizado.

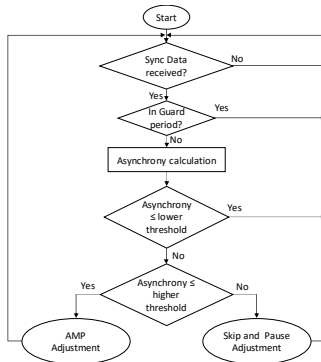


Fig. 5. Diagrama de flujo del mecanismo de sincronización implementado.

c) Configuraciones implementadas en los prototipos

Para los prototipos implementados, se han configurado los umbrales inferior y superior de la asincronía detectada en 60ms y 180ms, respectivamente. Se ha llegado a estos valores a través de múltiples pruebas en el laboratorio. Por un lado, 60ms es el umbral más bajo que garantiza el correcto funcionamiento obteniendo una precisión más que aceptable para el sistema videowall utilizando RPi como RD y, por otro lado, niveles de asincronía que sobrepasen los 180ms ya son perceptibles por los usuarios (p. ej., en contenido a 25fps, 200ms de asincronía implican un desfase de 5 fotogramas) y, por tanto, deben ser corregidos tan pronto como sea posible.

Adicionalmente, tras hacer también pruebas con diferentes valores, se ha configurado que se transmita la información temporal del M-RD cada 240ms y un periodo de guarda de 1s.

La técnica de ajuste suave de la reproducción (AMP) se ha configurado para que modifique la tasa de reproducción hasta un 10%, puesto que tras haber comprobado diferentes valores por debajo del 25% (límite para que se considere la variación de la tasa como imperceptible [29] [30]), es con el que mejores resultados se han obtenido, ya que permite ajustar la asincronía detectada de manera precisa y, a su vez, evita el vaciado del búfer del reproductor.

3) Validación visual preliminar

Como fase previa a las evaluaciones, y con el fin de comprobar el correcto funcionamiento del sistema propuesto y que, visualmente, el nivel de sincronización era aceptable, se realizaron test visuales con diferentes configuraciones de secciones en el prototipo B. La Fig. 6 muestra varios ejemplos: sección 3x2 con contenido almacenado (sup. izq.); sección 2x2 con contenido en vivo (canal Teledporte de RTVE) más sección 2x1 con contenido almacenado (sup. dch.); dos secciones 1x2 con contenido almacenado y una sección 2x1 con

contenido en vivo (inf. izq.); y en distribución disforme, dos secciones 2x2 y 2x1, con contenido almacenado (inf. dch.).



Fig. 6. Ejemplos de configuraciones de un sistema videowall 3x2.

B. Evaluación objetiva

Para llevar la evaluación objetiva en el prototipo A se han valorado dos casos diferentes: uno involucrando sólo contenido en vivo y otro sólo con contenido almacenado. Se ha comparado el nivel de sincronización alcanzado para contenido almacenado, utilizando el mecanismo de sincronización de terceros adoptado en [7] y el mecanismo propio descrito. Para el caso de contenido broadcast emitido en abierto, en España se recibe contenido con una resolución máxima de 1080p y a 25fps. Por tanto, la evaluación objetiva de contenido en vivo no exige a las RPi (RD) trabajar al límite de sus prestaciones. Es por ello que, con tal de probar el videowall exigiéndoles el máximo de sus prestaciones, se ha utilizado contenido 8K a 60fps (obtenido de Youtube⁴) para el caso de la evaluación con contenido almacenado. Dicho contenido 8K se ha troceado previamente en 16 regiones de 1080p, ya que el prototipo A sigue una distribución uniforme 4x4. En <https://bit.ly/2JKzh8T> se proporciona una muestra en vídeo del rendimiento del sistema y la precisión del nivel de sincronización alcanzado. Para calcular dicho nivel, se han realizado 10 sesiones de 5 minutos, durante las cuales cada RD ha registrado los valores del nivel de asincronía calculados. La Tabla II muestra los resultados de la asincronía media alcanzada y un intervalo de confianza (I.C.) del 95%, para ambos contenidos.

| Tipo de contenido | Valor medio (ms) | I.C. 95% (ms) |
|-------------------|------------------|---------------|
| Almacenado | 22 | ± 2.89 |
| En vivo | 41 | ± 1.81 |

El nivel de sincronización alcanzado resulta prácticamente imperceptible para el ojo humano (tal y como se ha comprobado en la evaluación subjetiva), puesto que, para ambos casos, la diferencia máxima de fotogramas con dichos valores de sincronización es de 2 fotogramas (1 fotograma a 60fps o a 25 fps implica ~17ms o 40ms, respectivamente).

Adicionalmente, se han registrado sesiones más largas para observar la evolución del nivel de asincronía. La Fig. 7a muestra los resultados registrados durante una sesión de aproximadamente 10 minutos, incluyendo, además, la evolución del nivel de sincronización utilizando el mecanismo de sincronización adoptado en [7]. Para una mejor interpretación, se ha añadido la media móvil (con grupos de 10 muestras). Se observan las fluctuaciones de los procesos de reproducción, aun cuando no se realizan ajustes de sincronización, y cómo el mecanismo que se implementa en este trabajo mejora el adoptado en [7]. En la Fig. 7b, se muestra la evolución de la sincronización registrada en el caso de la reproducción de contenido en vivo en uno de los S-RD

⁴ <https://youtu.be/ku3wCvIUdHQ> (último acceso: abril 2020)

involucrados durante una sesión de 20 minutos. Se pueden observar los ajustes realizados por el mecanismo propuesto, cada vez que el valor calculado supera el umbral inferior (60ms). En la Fig. 7c, se muestra la distribución de frecuencia acumulada para contenido almacenado (utilizando el mecanismo de sincronización adoptado en [7] y el propio) y contenido en vivo (solo con mecanismo propio, el adoptado en [7] no funciona para contenido en vivo). Se puede comprobar que se obtiene un mejor resultado para el caso de contenido almacenado con el mecanismo propio. Como era de esperar, se obtiene peor resultado cuando se reproduce contenido en vivo, pero sigue siendo satisfactorio, tal y como indica el nivel medio de sincronización obtenido (Tabla II). En la figura se puede comprobar, por ejemplo, que existe un 90% de probabilidad de tener un nivel de asincronía de 35ms con el mecanismo de sincronización propio en el caso de contenido almacenado. Adoptando el mecanismo adoptado en [7], para dicho porcentaje de probabilidad se obtendría una asincronía de 40ms. Respecto al contenido en vivo, la probabilidad de tener un valor de asincronía de 40ms, se reduce a un 70%.

C. Evaluación subjetiva

Durante la evaluación subjetiva, el prototipo B se ha ubicado en espacios públicos delante de cuatro servicios diferentes del Campus de Gandia de la UPV. Dichos servicios han sido el punto de información, secretaría del centro, la biblioteca y el servicio de deportes. En cada ubicación, ha sido utilizado durante una semana por trabajadores de cada servicio, con la finalidad de proporcionar al alumnado información relacionada con su servicio. En total, 15 trabajadores (sin ninguna relación directa con los autores ni con el proyecto) han participado en la evaluación. De ellos, el 62% eran mujeres, el 23% estaba entre los 30-39 años, el 54% entre los 40-49 años y el 23% entre los 50-59 años. Sólo un 30% de los participantes tenían un trabajo con un perfil técnico. Tras una sesión guiada de 30 minutos para la familiarización con el sistema, los participantes pudieron configurarlo y gestionarlo durante una semana, creando y modificando secciones, seleccionando los contenidos a reproducir, configurando y programando listas de reproducción, etc. Tras finalizar la semana, los participantes rellenaron un cuestionario sobre la usabilidad y respondieron a una serie de preguntas acerca del nivel de sincronización percibido. Se utilizó el conocido test de usabilidad SUS (*System Usability Scale* [31]), que proporciona una puntuación de 0 a 100 puntos, siendo 0 el peor resultado posible y 100 el mejor [32]. El test SUS consta de las siguientes 10 preguntas:

- Q1. *Creo que utilizaría este sistema con frecuencia.*
- Q2. *He encontrado el sistema innecesariamente complejo.*
- Q3. *Creo que el sistema era fácil de usar.*
- Q4. *Creo que necesitaría soporte técnico para utilizar el sistema.*
- Q5. *Las funcionalidades están bien integradas en el sistema.*
- Q6. *Creo que había demasiadas inconsistencias en el sistema.*
- Q7. *Imagino que la mayoría de las personas aprenderían rápidamente a utilizar este sistema.*
- Q8. *El sistema me ha parecido muy incómodo de usar.*
- Q9. *Me sentí muy seguro/a utilizando el sistema.*
- Q10. *Necesité aprender muchas cosas antes de poder utilizar este sistema.*

La Fig. 7d muestra los resultados obtenidos. La puntuación de la usabilidad media obtenida ha sido de 88.65, con una desviación estándar de 7.68. De acuerdo con [32], dicho valor implica que la usabilidad del sistema se puede considerar como *excelente*. Además, los participantes también valoraron el nivel

de sincronización percibido a partir de la escala *Mean Opinion Score* (MOS) [33]. Esta escala consta de 5 niveles, del 1 al 5, en la que el nivel 1 implica que el sistema que no está sincronizado en absoluto, y el nivel 5 implica que está perfectamente sincronizado. El valor medio obtenido ha sido de 4.3 puntos \pm 0.26 I.C. 95%. A pesar del buen resultado, en algunos casos se puede haber visto afectado por la etapa de sincronización inicial. Al iniciar la reproducción de un contenido existe un periodo transitorio hasta que cada uno de los S-RD alcanzan un nivel de sincronización aceptable.

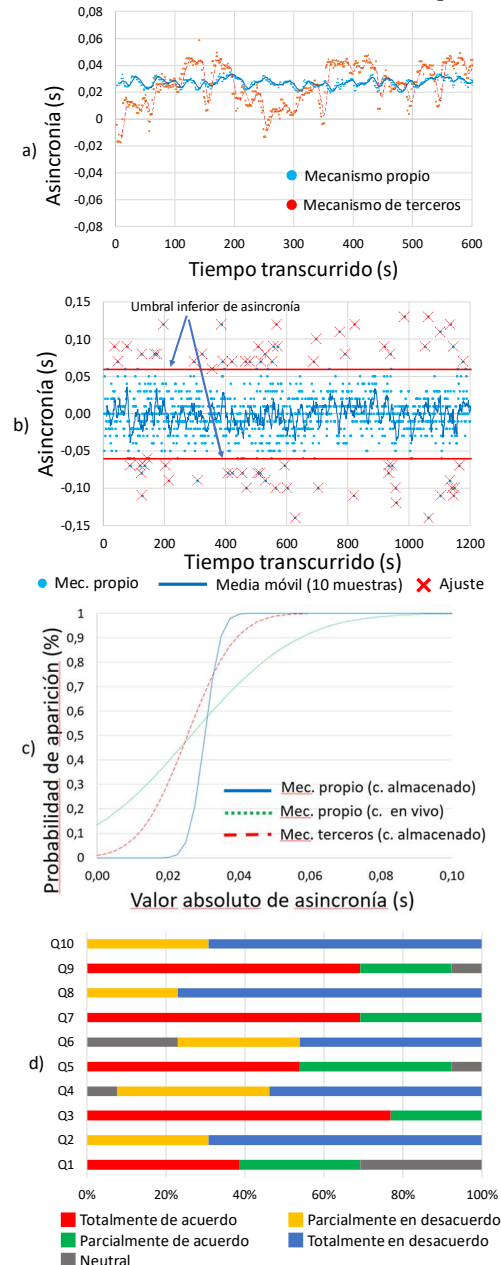


Fig. 7. Resultados obtenidos en evaluaciones objetiva (a, b, c) y subjetiva (d).

V. CONCLUSIONES

En este artículo se ha presentado y evaluado la arquitectura tanto HW como SW de un sistema videowall de bajo coste, escalable, configurable y personalizable a través de un servicio

dinámico de gestión y configuración basado en web. El sistema permite diferentes configuraciones y distribuciones de displays, así como la reproducción sincronizada tanto de contenido almacenado como de contenido en directo broadcast. Incluye un mecanismo de sincronización propio que ha sido validado de manera objetiva y subjetiva con resultados satisfactorios, mejorando la sincronización obtenida utilizando soluciones de terceros. Además, la usabilidad del sistema también ha sido validada subjetivamente como excelente [40].

Como trabajo futuro, se pretende implementar prototipos utilizando otros dispositivos de bajo coste más actualizados y potentes (p. ej. Raspberry Pi 4), para obtener un mejor rendimiento general y poder ofrecer visualización de contenidos de mayor resolución con un menor número de displays. También se aprovecharán los prototipos creados para evaluar otros esquemas de control de sincronización (SMS y DCS). Además, se estudiará, mediante el uso de multiplexores, la viabilidad de utilizar varias salidas de audio (p. ej. una por cada sección), característica interesante, p. ej., en videowalls en sitios públicos, donde se estén visualizando contenidos en diferentes secciones. Se redireccionarán las diferentes salidas de audio a diferentes grupos de altavoces (o interfaces de audio, tales como canales Bluetooth o radio). Así cada usuario podrá elegir la salida de audio que le interese en cada momento.

REFERENCIAS

- [1] T. Ni, et al., "A survey of large high-resolution display technologies, techniques, and applications," in *Proceedings - IEEE VR*, 2006, vol. 2006, p. 31, DOI: 10.1109/VR.2006.20.
- [2] L. P. R. Rivera, et al., "Visual data mining over a video wall," in *CONIELECOMP*, 2012, pp. 239–244, DOI: 10.1109/CONIELECOMP.2012.6189916.
- [3] J. M. E. M. Van Der Werf, et al., "Facilitating collaborative decision making with the software architecture video wall," in *Proceedings - IEEE, ICSAW 2017: Side Track Proceedings*, 2017, pp. 137–140, DOI: 10.1109/ICSAW.2017.27.
- [4] P. Yin, et al., "Multi-screen tiled displayed, parallel rendering system for a large terrain dataset," *IJVR*, vol. 5, no. 4, pp. 47–54, 2006.
- [5] S. Noda, et al., "Implementation of High Presence Video Communication System for Multiple Users Using Tiled Display Environment," in *Proceedings - IEEE WAINA*, 2015, pp. 494–499, DOI: 10.1109/WAINA.2015.81.
- [6] R. Bundulis and G. Arnicans, "Architectural and technological issues in the field of multiple monitor display technologies," in *FALA*, 2013, vol. 249, pp. 317–329, DOI: 10.3233/978-1-61499-161-8-317.
- [7] P. Salvador, et al., "Sistema videowall de bajo coste basado en Raspberry Pi, personalizable y configurable dinámica y remotamente vía Web," in *JITEL 17*, 2017, pp. 318–325, DOI: 10.4995/jitel2017.2017.6619.
- [8] M. Montagud, et al., "How to perform AMP? Cubic adjustments for improving the QoE," *Comput. Commun.*, vol. 103, pp. 61–73, May 2017, DOI: 10.1016/j.comcom.2017.01.017.
- [9] L. Renambot, et al., "SAGE: the Scalable Adaptive Graphics Environment," in *WACE*, 2004.
- [10] T. Marrinan et al., "SAGE2: A new approach for data intensive collaboration using Scalable Resolution Shared Displays," in *CollaborateCom 2014 - Proceedings of the 10th IEEE CollaborateCom*, 2015, pp. 177–186, DOI: 10.4108/icst.collaboratecom.2014.257337.
- [11] K. U. Doerr and F. Kuester, "CGLX: A scalable, high-performance visualization framework for networked display environments," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 3, pp. 320–332, 2011, DOI: 10.1109/TVCG.2010.59.
- [12] G. Humphreys et al., "Chromium: A stream-processing framework for interactive rendering on clusters," in *Proceedings of the SIGGRAPH '02*, 2002, pp. 693–702, DOI: 10.1145/566570.566639.
- [13] G. P. Johnson, et al., "DisplayCluster: An interactive visualization environment for tiled displays," in *Proceedings - IEEE CLUSTER 2012*, 2012, pp. 239–247, DOI: 10.1109/CLUSTER.2012.78.
- [14] R. Bundulis and G. Arnicans, "Infiniviz: Taking Quake 3 Arena on a Large-Scale Display System to the Next Level," in *FRUCT*, 2018, vol. 2018-Novem, pp. 91–98, DOI: 10.23919/FRUCT.2018.8588078.
- [15] F. Boronat, et al., "Multimedia group and inter-stream synchronization techniques: A comparative study," *Inf. Syst.*, vol. 34, no. 1, pp. 108–131, Mar. 2009, DOI: 10.1016/j.is.2008.05.001.
- [16] M. Montagud, et al., "Inter-destination multimedia synchronization: Schemes, use cases and standardization," *Multimed. Syst.*, vol. 18, no. 6, pp. 459–482, Nov. 2012, DOI: 10.1007/s00530-012-0278-9.
- [17] K. Y. Kim, et al., "The design of server virtualization based video-wall control system," in *Proceedings ICCCT*, 2012, pp. 401–404.
- [18] D. C. A. Bulterman, "SMIL: Synchronized multimedia integration language," in *MediaSync: Handbook on Multimedia Synchronization*, 2018, pp. 359–385.
- [19] R. Bundulis and G. Arnicans, "Concept of virtual machine based high resolution display wall," in *2014 IEEE 2nd Workshop AIEEE - Proceedings*, 2014, vol. 21, DOI: 10.1109/AIEEE.2014.7020317.
- [20] A. Jimenez and D. Rincon, "Videowall disforme sobre redes IP," *UPCommons*, pp. 1–77, 2016.
- [21] "PiWall," <http://www.piwall.co.uk> (último acceso: abril 2020).
- [22] B. A. Alabdulsalam, et al., "Dynamic Video Wall Tile Creation Using Raspberry Pi3," in *Proceedings - IEEE 13th ISADS*, 2017, pp. 268–271, DOI: 10.1109/ISADS.2017.45.
- [23] V. Mihalopoulos and K. Sveronis, "Creating a videowall with yodeck." [Online]. Available: <https://yodeck.com/docs/display/YO/Creating+a+Video-Wall+with+Yodeck> (último acceso: abril 2020).
- [24] I. H. Shin, et al., "S/W Based frame-level synchronization for irregular screen processing system," *ETRI J.*, vol. 38, no. 5, pp. 868–878, Oct. 2016, DOI: 10.4218/etrij.16.2616.0006.
- [25] J. Kanda, et al., "A multiple terminal synchronous display method for ultra-high resolution display systems," in *Digest of Technical Papers - IEEE ICCE*, 2014, pp. 510–511, DOI: 10.1109/ICCE.2014.6776109.
- [26] D. Mills, et al., "RFC 5905 - Network Time Protocol," *IETF* 2010.
- [27] H. Schulzrinne, et al., "RFC 3550 - RTP," *IETF* 2003.
- [28] I. Fette, et al., "RFC6455 - The WebSocket Protocol," *IETF* 2011.
- [29] H. C. Chuang, et al., "Content-aware adaptive media playout controls for wireless video streaming," *IEEE Trans. Multimed.*, vol. 9, no. 6, pp. 1273–1283, Oct. 2007, DOI: 10.1109/TMM.2007.902884.
- [30] Y. F. Su, et al., "Smooth control of adaptive media playout for video streaming," *IEEE Trans. Multimed.*, vol. 11, no. 7, pp. 1331–1339, Nov. 2009, DOI: 10.1109/TMM.2009.2030543.
- [31] J. Brooke, "SUS-A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, no. 194, pp. 4–7, 1996.
- [32] A. Bangor, et al., "Determining what individual SUS scores mean: adding an adjective rating scale," *J. Usability Stud.*, vol. 4, no. 3, pp. 114–123, 2009.
- [33] International Telecommunication Union, "P.800: Methods for subjective determination of transmission quality," *ITU-T Recomm.*, vol. 800, 1996.



Dani Marfil estudió Ingeniería Técnica Informática de Sistemas (2011), Grado de Sistemas de Telecomunicación, Imagen y Sonido (2015) y Máster en Tecnologías, Redes y Sistemas de Telecomunicación (2016) en la Universitat Politècnica de València (UPV). Actualmente, está realizando el Doctorado y forma parte del grupo de investigación IIM R&D (<http://iim.webs.upv.es>). Es autor de un capítulo de libro y varios artículos.



Fernando Boronat (M'93–SM'11), estudió Ingeniería de Telecomunicación en la UPV. Recibió el grado de Doctor en 2004 y actualmente es Profesor Asociado en el Departamento de Comunicaciones de la UPV. Es el responsable del grupo IIM R&D. Sus principales áreas de interés son los sistemas, protocolos y sincronización multimedia. Es autor de varios libros y capítulos de libros, y de más de 100 artículos de investigación.



Juan González estudió el Grado de Sistemas de Telecomunicación, Imagen y Sonido (2016) y el Máster en Bioinformática y Bioestadística en la Universitat Oberta de Catalunya y la Universitat de Barcelona (2018). Es miembro de IIM R&D. Sus áreas de interés son el desarrollo de aplicaciones web y móviles, RV/RA y técnicas de machine learning.



Anna Vidal es doctora en Matemáticas (1999) y Profesor Asociado en la UPV. Forma parte del grupo IIM R&D centrándose su investigación en el diseño, modelado y análisis de técnicas de sincronización multimedia.