UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

INGENIERÍA EN ELECTRÓNICA

Trabajo Fin de Carrera

Diseño, implementación y evaluación de un sistema de localización de locutores basado en fusión audiovisual

María Cabello Aguilar
30 de Septiembre del 2010

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

INGENIERÍA EN ELECTRÓNICA

Trabajo Fin de Carrera

Diseño, implementación y evaluación de un sistema de localización de locutores basado en fusión audiovisual

Alumno: María Cabello aguilar

Director: Javier Macías Guarasa

Tribunal:

Presidente: D. Daniel Pizarro Pérez .

Vocal 1º: D. Juan Carlos García García.

Vocal 2º: D. Javier Macías Guarasa.

Calificación: .....

Fecha: .....

Agradecimientos

A mis padres, que me han dado su apoyo todos los días de mi vida, en los buenos y malos momentos, y sin los que no habría sido capaz de superar ciertas etapas. Gracias por sentiros siempre orgullosos de mí.

A mi hermano Joaquín, el gran pilar de mi vida que aunque tengo lejos siento siempre conmigo.

A Víctor, por estar día tras día a mi lado, aguantando mis peores momentos y ayudándome en este largo trabajo.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Resumen

1.1. Resumen

Este proyecto describe el diseño, implementación y evaluación de un sistema de localización de locutores basado en fusión audiovisual.

Se han desarrollado las siguientes tareas:

- Mejora del sistema de experimentación basado en información acústica, llegando a ser más flexible, eficiente y robusto; y permitiendo la realización de los experimentos de una manera automática.
- Mejora del sistema de localización de hablantes basado en información acústica añadiendo nuevas funcionalidades.
- Mejora del sistema de visión para la localización de objetos, obteniendo un software más versátil con nuevas y mejoradas funcionalidades.
- Diseño e implementación de un sistema basado en fusión audiovisual que permite la localización de objetos partiendo de información acústica y procedente de visión.

Palabras clave: Sistema de localización de hablantes, Espacios Inteligentes, sistema de detección de objetos, fusión audiovisual, bases de datos CHIL.

Capítulo 2

Abstract

2.1. Abstract

This project describes the design, implementation and evaluation of a speaker location system based on audiovisual fusion.

The following tasks have been developed:

- Improved experimental system based on acoustic information, becoming more flexible, efficient and robust, allowing the realization of experiments in an automatic way.
- Improved tracking system based on acoustic information speakers adding new features.
- Improved vision system for object location, obtaining a more versatile software with new and enhanced functionalities.
- Design and implementation of a system based on audiovisual fusion that allows the location of objects using information from audio and vision.

Keywords: Speaker location system, Intelligent Spaces, objects detection system, audiovisual fusion, CHIL databases.

Capítulo 3

Introducción

3.1. Presentación

El análisis automático de Espacios Inteligentes a partir del procesamiento de múltiples sensores es un área de cada vez mayor actividad científica.

En este contexto, las tareas de detección, localización y seguimiento de personas son fundamentales para mejorar los procesos de interacción con el entorno, o con otras personas u objetos del mismo [1]. Las áreas de explotación de dichas tareas abarcan tanto aspectos ligados al procesamiento de señal (por ejemplo técnicas de mejora de la señal de habla captada por micrófonos lejanos [2] [3], dada la fuerte sensibilidad de la misma a los problemas de rerveberación, ruido aditivo y baja relación señal ruido [4] [5] o técnicas de indentificación de locutores y de detección de eventos acústicos localizados), como aquellos relacionados con el análisis de las interacciones humanas dentro del entorno, y de los humanos con otros elementos (por ejemplo robots móviles [6]).

El Grupo de Ingeniería Electrónica aplicada a Espacios Inteligentes y Transporte del Departamento de Electrónica de la Universidad de Alcalá ha arrancado una línea de actividad en la que se plantean trabajos orientados a la explotación conjunta (fusión) de la información acústica generada por hablantes y la procedente de capturas de vídeo del entorno, para mejorar la interacción de éstos en Espacios Inteligentes, una de cuyas primeras aplicaciones será la localización robusta de locutores.

El trabajo que aquí se propone representa el primer paso en esta línea, orientado fundamentalmente a diseñar, implementar y evaluar un sistema de fusión de información acústica y visual para tareas de localización y seguimiento de hablantes en un Espacio Inteligente.

En este Proyecto Fin de Carrera se parte de trabajos iniciados por los Proyectos Fin de Carrera de Eva Muñoz Herraiz [7] (“Diseño, implementación y evaluación de técnicas de localización de fuente y de mejora de la señal de habla en entornos acústicos reverberantes: aplicación a sistemas de reconocimiento automático de habla”), Carlos Castro González [8] (“Speaker Localization Techniques in Reverberant Acoustic Environments) y María Cabello Aguilar [9] (”Comparativa teórica y empírica de métodos de estimación de la posición de múltiples objetos“), y especialmente de la Tesis Doctoral de Marta Marrón Romera [10] (”Seguimiento de múltiples objetos en entornos interiores muy poblados basado en la combinación de métodos probabilísticos y determinísticos“).

3.2. Motivación y objetivos del proyecto

Los objetivos del proyecto son:

- Mejorar las prestaciones de las herramientas y algorítima disponibles en el Grupo en sistemas de procesamiento de audio para tareas de localización y seguimiento de locutores (sistemas de ayuda a la experimentación, etc.). En concreto se abordan trabajos orientados a:
    - Mejorar las prestaciones del sistema de localización acústica de partida para incorporarle facilidades de procesamiento de ficheros de audio multicanal y de procesamiento de múltiples agrupaciones de micrófonos de forma integrada o independiente.
    - Disponer de un sistema de experimentación versátil que facilite la realización sistemática de experimentos sobre múltiples bases de datos (incluyendo tareas de configuración, ejecución y evaluación).

- Diseñar e implentar un sistema de localización de hablantes combinando la información acústica procedente de múltiples agrupaciones de micrófonos con la información visual capturada con múltiples cámaras en un Espacio Inteligente (Figura 3.1), siguiendo el esquema de bloques mostrado en la Figura 3.2.

  Los requisitos que cumple el trabajo son los siguientes:

  - Incorporar los procesos que procedan en los sistemas de localización y seguimiento de múltiples locutores, existentes o en desarrollo dentro del Grupo, con vistas a la mejora de las tasas de fiabilidad obtenidas.
  - Ser flexible en el sentido de permitir modificar con facilidad los parámetros de control disponibles en los algoritmos de estimación utilizados.
  - Ser flexible en el sentido de permitir la cómoda incorporación y control de nuevos algoritmos de estimación de fiabilidad en localización y seguimiento.
  - Estar bien documentado para facilitar su utilización en futuros proyectos.
  - Disponer de un software eficiente y robusto.

- Evaluar los algoritmos de localización y seguimiento implementados, realizando experimentos utilizando el software desarrollado y las bases de datos multimodales disponibles en el Grupo. La evaluación cumple los siguientes requisitos:

  - Medir las prestaciones de los sistemas de localización y seguimiento utilizando únicamente información acústica, visual y la combinación de ambas. Se utilizan las estrategias de evaluación y métricas de calidad propuestas dentro del proyecto CHIL [11] (en la evaluación CLEAR).
  - Medir las prestaciones de las técnicas de localización y seguimiento implementadas en diferentes condiciones acústicas y visuales reales (en función de las bases de datos disponibles).
  - Buscar conclusiones razonadas sobre la validez de los resultados obtenidos con las técnicas implementadas.
  - Interpretar los resultados obtenidos a la vista de su fiabilidad estadística, considerando en su justa medida las mejoras o degradaciones observadas (respecto a los sistemas de partida).

3.3. Organización de la memoria

Este proyecto está formado por un total de nueve capítulos, cuyos contenidos se detallan a continuación:

- **Capítulo 1** - *Resumen* - En este capítulo se explica de forma muy breve y concisa las tareas abordadas por el actual trabajo.
- **Capítulo 2** - *Abstract* - Resumen del trabajo escrito en inglés.
- **Capítulo 3** - *Introducción*. Capítulo actual en el que se plantean los objetivos principales y secundarios del proyecto, se explican los puntos de partida en los que se ha basado todo el trabajo para terminar con una explicación sobre la organización de la memoria.

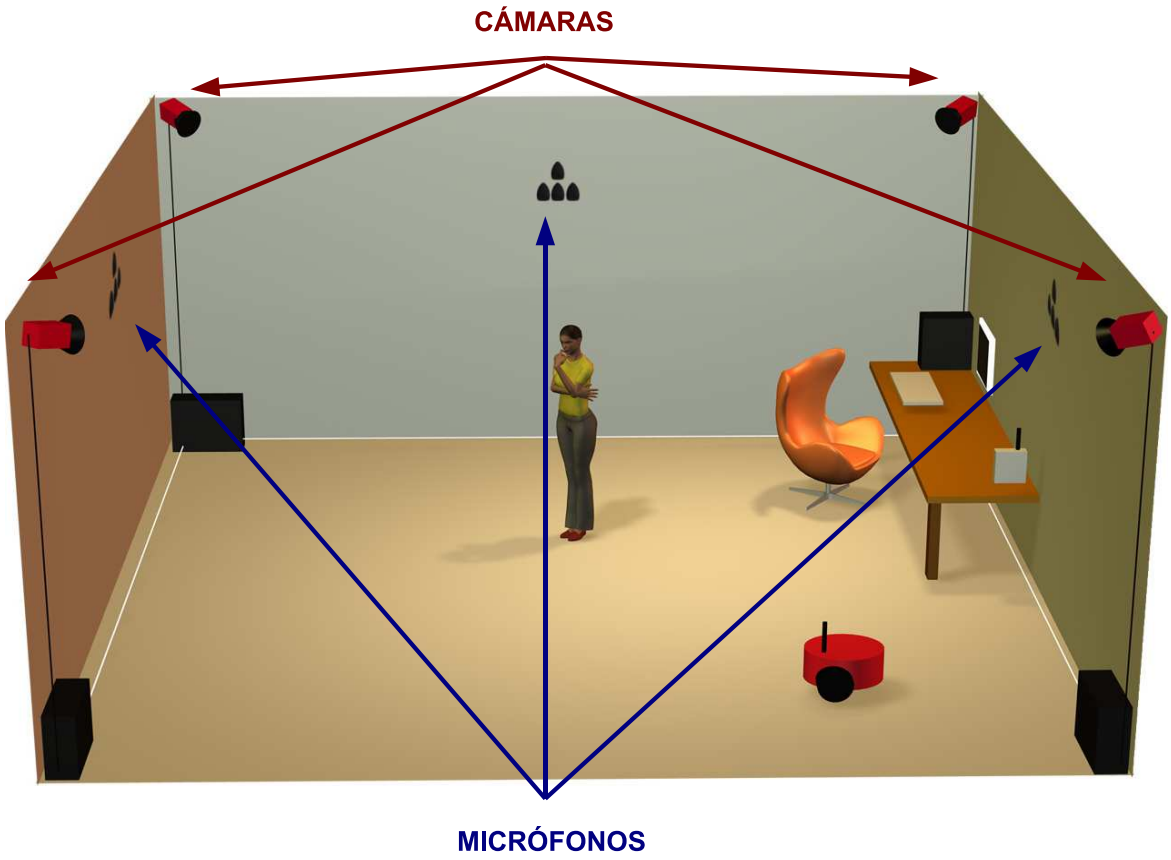


Figura 3.1: Espacio Inteligente

- **Capítulo 4** - *Estudio teórico*. Se detallan todos aquellos conceptos teóricos necesarios para comprender la funcionalidad final de los desarrollos implementados, así como aquella base teórica en la que se basan. Se exponen conceptos teóricos sobre la localización basada en audio y la localización basada en vídeo.
- **Capítulo 5** - *Desarrollo algorítmico y herramientas*. En este capítulo se exponen aquellas mejoras desarrolladas tanto en el sistema de localización acústica como en el sistema de experimentación basado en fusión audiovisual.
- **Capítulo 6** - *Resultados experimentales*. Se muestran los resultados obtenidos mediante la aplicación de los desarrollos implementados, tanto en forma de tablas como de forma gráfica.
- **Capítulo 7** - *Conclusiones y trabajos futuros*. Se plantean las conclusiones obtenidas tras la finalización del trabajo, así como una serie de trabajos que pueden ser ejecutados en el futuro y que son de interés.
- **Capítulo 8** - *Manual de usuario*. Manual completo del sistema de experimentación basado en audio y de aquel basado en fusión audiovisual, cuyo fin es permitir al usuario replicar el funcionamiento de los sistemas empleados y desarrollados en el presente trabajo.
- **Capítulo 9** - *Apéndices*. Información adicional y de interés relacionada con el trabajo.
- **Bibliografía** - Información consultada para la redacción y elaboración de esta memoria.

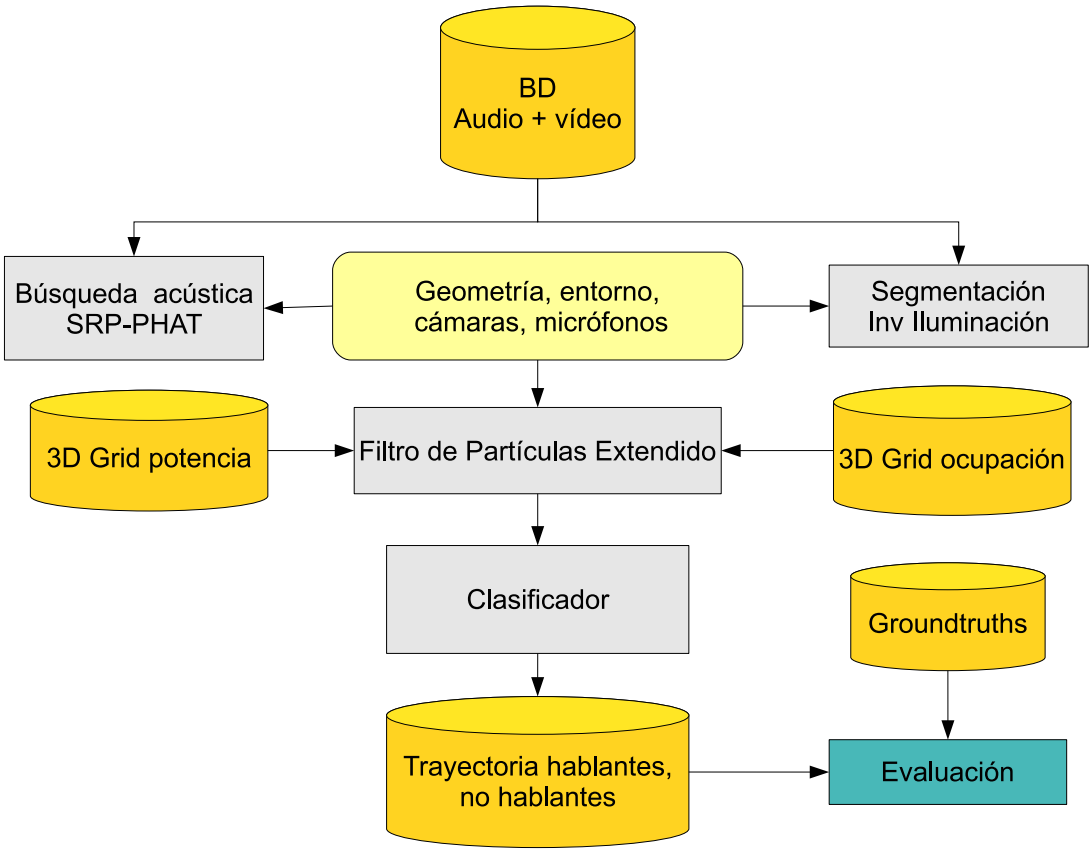


Figura 3.2: Arquitectura del sistema de fusión audiovisual

Capítulo 4

Estudio teórico

4.1. Introducción

Este trabajo se centra en el campo de los "Espacios Inteligentes", término adoptado recientemente que se enmarca dentro de una tendencia de investigación más amplia, la "computación ubicua". La computación ubicua plantea un espacio dotado de un conjunto de sistemas sensoriales, de comunicación y de cómputo inteligente, que son imperceptibles para el usuario y se encuentran constantemente recogiendo información del entorno y cooperando entre ellos para ofrecer la ayuda necesaria a cada persona. Este tipo de entornos reciben el nombre de Espacios Inteligentes, que requieren de una red de sensores capaz de obtener información relevante del entorno, y a partir de esos datos especificar el comportamiento que se debe proporcionar en función de las necesidades de los usuarios [12].

Existen gran cantidad de trabajos realizados con el objetivo de localizar determinados objetos o *targets* dentro de una escena bajo análisis empleando información procedente de un único tipo de sensores; como por ejemplo videocámaras [13] [14], arrays de micrófonos [15] [8], infrarrojos, etc. Cada tipo de sensores posee una serie de debilidades y fortalezas, por lo que se ha planteado la utilización combinada de varios de ellos como forma de aumentar la robustez de los algoritmos [16].

Cuando se llevan a cabo videoconferencias "inteligentes", en aplicaciones de interfaz persona-máquina, análisis automático de escenas, vigilancia, etc., en las cuales se ven implicadas fuentes sonoras, la utilización de esta información resulta interesante a la hora de la detección, localización y seguimiento (se usará tanto el término español como el inglés *tracking*) de determinados objetivos (de la misma manera se empleará indistintamente el término en español o en inglés *targets*) dentro de la misma. Además hay que tener en cuenta que el habla es la forma natural de comunicarse de las personas, por lo que resulta aconsejable su utilización para mejorar la interacción persona-máquina, y por lo tanto su uso en los Espacios Inteligentes.

Sin embargo, la localización empleando técnicas acústicas sólo ofrece buenos resultados cuando se trabaja con un conjunto de micrófonos de características determinadas, siendo además la estimación ofrecida poco precisa en casos en los que se busca precisión milimétrica. En el momento en el que se trabaja con una baja relación señal-ruido (SNR) o en espacios muy reverberantes, estos algoritmos no son capaces de ofrecer una buena estimación de la posición. Por este motivo es necesario incluir sensores adicionales como aquellos que trabajan con información procedente de visión.

Los sensores de vídeo constituyen una alternativa complementaria debida principalmente a la gran cantidad de información que ofrecen, otorgando esto potencia de precisión y robustez superior a la de los sensores de audio, si bien la complejidad de los algoritmos es superior.

4.2. Localización basada en información acústica

4.2.1. Introducción

En esta sección se van a explicar aquellos conceptos teóricos necesarios para la correcta comprensión de las técnicas de reconocimiento de voz en espacios reverberantes empleando para tal arrays de micrófonos, y empleadas en este trabajo.

Es de destacar que existen diversos factores que empeoran considerablemente los resultados ofrecidos por los algoritmos de procesamiento de habla, como son el ruido o la reverberación. En el caso de emplear en el reconocimiento micrófonos cercanos a las personas, la influencia de estos factores es poco importante y permite el desarrollo de exitosos sistemas de comunicación persona-máquina. En cambio, cuando las señales son recibidas por un conjunto de micrófonos alejados de la fuente de voz, la influencia de estos factores degrada en gran medida las prestaciones del sistema. Para paliar estos fectos se emplean técnicas que son descritas posteriormente.

4.2.2. Estado del arte en la detección, localización y estimación de pose con sensores de audio

El principal objetivo de un sistema de localización es soneguir una gran precisión, que en el caso de llevarse a cabo mediante el empleo de arrays de micrófonos depende principalmente de cuatro factores:

1. Cantidad y calidad de los micrófonos empleados.
2. Ubicación relativa de los micrófonos entre sí y con respecto a la fuente sonora.
3. Nivel de ruido y reverberación del entorno.
4. Número de fuentes activas y contenido espectral de sus emisiones.

Por tanto, un sistema basado en audio debería ser capaz de estimar el número de hablantes activos en cada instante de tiempo (detección), su posición instantánea (localización), su trayectoria espacio-tiempo (*tracking*) y su orientación (pose). Para ello se definen tres tipos de estrategias diferentes, no siendo necesariamente excluyentes [17]:

- Diferencias en los tiempos de vuelo de las ondas acústicas desde los *targets* activos hasta los micrófonos. Esta opción es la más utilizada por su versatilidad y aplicabilidad práctica con costes de proceso asumibles [18].
- Empleo de la respuesta al impulso, de difícil medida y calibración y aún en etapa de desarrollo, aunque se pueden encontrar resultados en trabajos al respecto en [19].
- Uso de la información de micrófonos direccionales orientados hacia diferentes direcciones del espacio [20], calculando las posiciones en base a la función de transferencia de cada uno de los micrófonos.

En este trabajo se aborda la primera de las soluciones expuestas, al comprender aquella que mejores resultados ofrece en situaciones realistas.

Se describen a continuación las alternativas existentes para las fases de detección, localización y estimación de pose:

- Detección:
  La señal de audio es por naturaleza intermitente, y por lo tanto en los momentos de silencio no se deben realizar estimaciones. Existen detectores de actividad de voz o Voice Activity Detectors (VAD) que emplean características individuales del canal para calcular las métricas necesarias, combinadas con reglas de clasificación basadas en umbrales fijos

o recalculados en periodos de silencio. Estos métodos suponen problemas en espacios con baja SNR y especialmente con ruidos no estacionarios.

Existen otros métodos alternativos que explotan la distribución espacial de múltiples micrófonos basados en el cálculo de actividad acústica por sectores, y que pueden encontrarse en [15], o estrategias como la presentada en [21] basada en la Cross-Power Spectrum (CSP) de la señal capturada. Sin embargo, una de las alternativas más prometedoras es la abordada por investigadores de IDIAP en [22], y denominada *short-term clustering*, la cual no necesita saber el número de *targets* y en la que se aplican técnicas de agrupación online no supervisada.

- Localización:
  En cuanto al problema de localización, los algoritmos existentes pueden ser clasificados en tres grandes grupos [23]:
  - Los que emplean *Time Differences of Arrival* (TDOA) [24] [25], que se explican con detalle en la sección 4.2.5.
  - Los basados en *High-resolution Spectral Estimation*, que se centran en la explotación de las propiedades de la *Cross-Sensor (spatial) Covariance Matrix (CSCM)* del array, a partir de la cual son capaces de dividirla en dos subespacios, uno que contiene la señal de habla y otro la de ruido [26]. Su principal inconveniente es que no son implementables cuando el número de llegadas incorreladas supera al número de sensores del array (condición típica en entornos realistas).
  - Los que utilizan *Steered Response Power* (SRP), detallados en la sección 4.2.5.
- Estimación de pose:
  En el contexto de los arrays de micrófonos, la estimación de la pose se refiere a la obtención de la dirección hacia la que está enfocada la fuente sonora. En estas líneas existen varios trabajos como [27], en el que se toma como base un algoritmo SRP-PHAT y se realiza la búsqueda de la orientación de los *targets* estableciendo una ponderación de la contribución de cada par de micrófonos para las orientaciones posibles. Una aproximación similar se presenta en [28]. Aproximaciones adicionales se basan en consdieraciones de la directividad del hablante [29].

Actualmente existe una necesidad de minimizar el número de micrófonos empleados, para lo que surge el concepto de procesamiento de habla binaural, no sólo para tareas de reconocimiento automático del habla [30], sino también en aplicaciones de localización [31] [32] [33]. La problemática asociada al procesamiento de habla binaural radica en la imposibilidad de aplicar las estrategias clásicas basadas en la estimación de retardos, dado que los micrófonos están enfrentados y la configuración de la cabeza y el torso del locutor afectan a la respuesta de ambos de una forma compleja [34]. Este tipo de sistemas ha recibido relativamente poca atención en la literatura dado el elevado coste de los equipos de calidad que permiten abordar investigaciones con garantías de éxito.

4.2.3. Propagación de las ondas sonoras

Las ondas sonoras se propagan a lo largo de los fluidos como ondas longitudinales. Empleando las ecuaciones de Newton del movimiento para considerar un volumen infinitesimal de un fluido se desarrolla una ecuación que modela la propagación de la onda. Una ecuación generalizada para ondas acústicas es compleja de obtener ya que depende de propiedades del propio fluido. A

lo largo de este trabajo se asume en todo momento que el sonido se transmite en ondas esféricas, como se expone en la ecuación lineal de la onda [23]. Existen modelos más complejos pero que no se han empleado en el desarrollo del algoritmo de estimación de la localización de los hablantes utilizado en el sistema de experimentación de este trabajo.

A continuación se pasa a explicar cómo afecta la propagación de la onda a la señal recibida por un micrófono en un entorno definido como una sala pequeña. Existen 3 factores influyentes que afectan a la propagación de la señal de voz: atenuación, ruido y reverberación.

4.2.3.1. Efecto de la atenuación, ruido y reverberación

En la propagación de las ondas esféricas, la amplitud de la señal decrece proporcionalmente con la distancia recorrida, como se describe en [23]. Como consecuencia de ello, la SNR decae de la misma manera, pudiéndose dar el caso de que la señal se encuentre por debajo del nivel de ruido si los micrófonos están en posiciones lejanas a la fuente de dicha señal.

Además, en cualquier situación aparece un ruido acústico que se suma a la señal. Como se explica en [29] se considera ruido a cualquier aportación externa no deseada en la señal acústica y captada por el micrófono. El ruido se puede clasificar en dos tipos fundamentalmente. En primer lugar se encuentra el no direccional referido al llamado ruido "de fondo". Este tipo de ruido reduce la SNR pero no influye de forma importante en la componente principal de la señal, aquella perteneciente al camino directo de la fuente acústica dominante. Por otro lado, el direccional (por ejemplo otra persona hablando en la misma habitación o ruidos procedentes de áreas concretas) introduce ambigüedad en la estimación de las posiciones ya que es una aportación de elevada importancia.

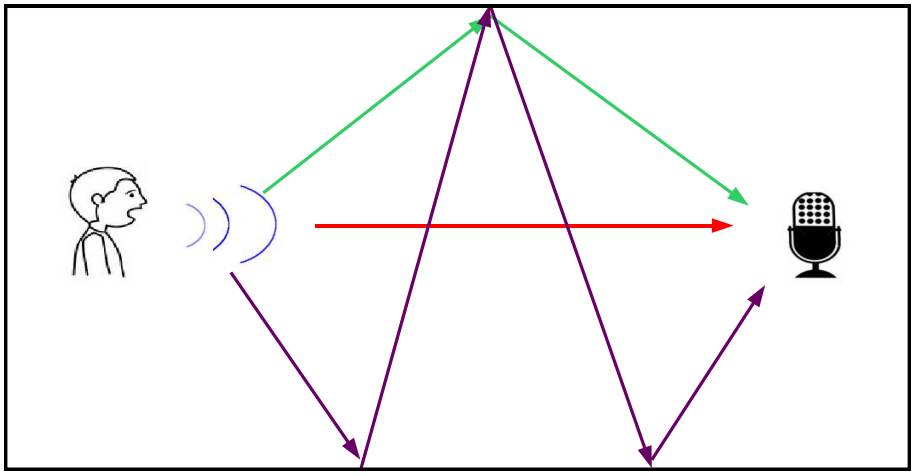


Figura 4.1: Propagación de una onda en una sala

Por último, comentar que en los espacios cerrados existe reverberación, fenómeno por el cual una onda sonora alcanza su objetivo a través de múltiples caminos, como se observa en la Figura 4.1. Como consecuencia de ello, la señal recibida en el micrófono contiene una aportación de la señal fuente original y numerosas copias de ella retardadas, atenuadas y distorsionadas; debido a la reflexión y difracción de las ondas. Esta reverberación se ve influenciada por dos factores, como son el tamaño de la sala o habitacion y la estructura de la misma. Este efecto comentado influye en la estimación de la localización de los hablantes de forma negativa, dando lugar a estimaciones erróneas.

4.2.3.2. Propagación del camino directo

Bajo condiciones acústicas simples [35] y en un espacio de libre propagación, donde no existen objetos que interfieran a las ondas, la señal recibida en un micrófono está relacionada linealmente con la señal original como se expresa en la ecuación 4.1. Esta afirmación no se cumple en los entornos donde se ha llevado a cabo la experimentación, pero sirve para caracterizar la propagación por el camino directo del sonido de la fuente al receptor, incluso en presencia de reverberación.

$$x_{direct}(r,t) = \frac{a}{r} \cdot s(t - \frac{r}{c}) = \frac{a}{r} \cdot s(t - \tau) \tag{4.1}$$

donde $x_{direct}(r,t)$ es la señal capturada por el micrófono y correspondiente al camino directo, $s(t - \frac{r}{c})$ es la señal original, a es la amplitud de la onda sonora, r es la distancia desde la fuente al receptor, c es la velocidad del sonido y τ el retardo de tiempo entre transmisor y receptor.

4.2.3.3. Modelo de la señal del micrófono

La siguiente ecuación representa la señal recibida por un micrófono m, procedente de una fuente sonora en un entorno reverberante. Mediante esta ecuación queda representada la componente de la señal correspondiente al camino directo y aquellas procedentes del efecto de la reverberación:

$$x_m(t) \simeq \frac{a}{r_m} \cdot s(t - \tau_m) + s(t) * u_m(\overrightarrow{r}_s, t) + v_m(t) \tag{4.2}$$

donde r_m es la distancia desde la fuente al receptor m, τ_m es el tiempo de retardo entre el transmisor y el receptor m, u_m representa la respuesta al impulso caracterizando todos los caminos acústicos excepto el camino directo, $\overrightarrow{r_s}$ es la localización de la fuente y $v_m(t)$ es algún tipo de ruido aditivo.

Se denomina $\widetilde{v}_m(t)$ a un nuevo término del ruido que incluye el ruido reverberante más el acústico:

$$\widetilde{v}_m(t) = s(t) * u_m(\overrightarrow{r}_s, t) + v_m(t) \tag{4.3}$$

Quedando entonces la ecuación de la señal del micrófono:

$$x_m(t) \simeq \frac{a}{r_m} \cdot s(t - \tau_m) + \widetilde{v}_m(t) \tag{4.4}$$

En la mayor parte de los casos sólo se tiene en cuenta la componente del camino directo de la señal, sin embargo, es conveniente emplear este modelo en el que se añade una componente de ruido (ruido acústico y reverberación) y se representa la señal como la suma de varias componentes que son copias escaladas y retardadas de la señal origen.

4.2.3.4. Campo lejano y campo cercano

Como se comentó anteriormente, el sonido se propaga mediante ondas esféricas. Sin embargo, cuando la distancia entre la fuente de la onda y el receptor r es mucho mayor que la longitud física del receptor R (distancia entre el primer y el último micrófono *d(N-1)*, donde d es la

distancia entre micrófonos y N el número de elementos), se puede aproximar afirmando que las ondas sonoras parecen planas al ser su curvatura demasiado pequeña en relación a la longitud del receptor. A este aspecto se le denomina condición de *campo lejano*, que se caracteriza por la siguiente relación:

$$|r| > \frac{2R^2}{\lambda} \tag{4.5}$$

Cuando no se cumple esta inecuación se habla de condición de *campo cercano*. Este aspecto es importante en el momento de calcular la diferencia de tiempo entre las señales de dos micrófonos.

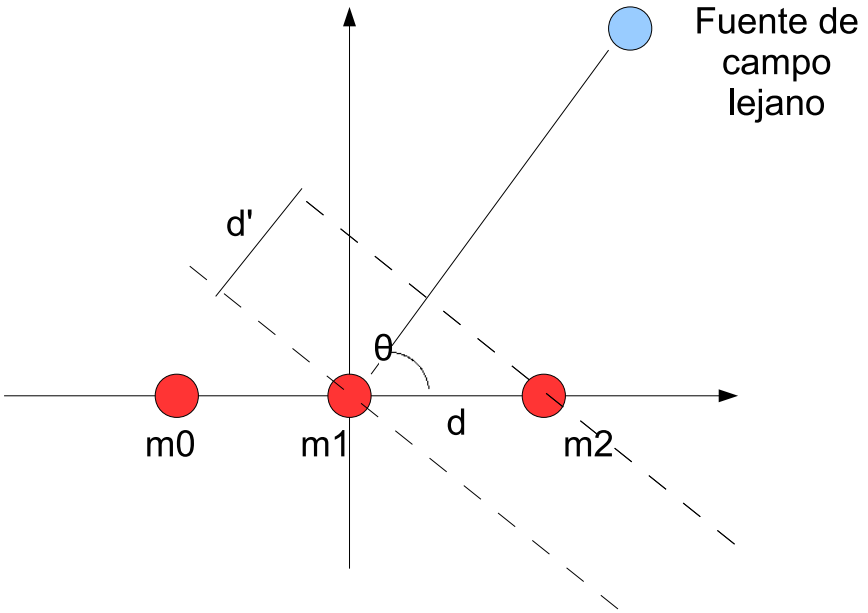


Figura 4.2: Propagación de una onda en una situación de campo lejano

En el ejemplo de campo lejano mostrado en la figura 4.2, la señal de voz tarda τ_0 segundos en alcanzar el micrófono *m1* y τ_1 segundos en alcanzar el micrófono *m2*. A simple vista se observa que la señal necesita recorrer una distancia extra *d'* para alcanzar el micrófono *m1*. Entonces se puede afirmar que el retardo de tiempo entre las señales de los dos micrófonos es la siguiente:

$$\Delta\tau = \tau_0 - \tau_1 = \frac{d'}{c} = \frac{d \cdot \cos\theta}{c} \tag{4.6}$$

donde c es la velocidad del sonido y θ es la dirección de llegada de la señal al conjunto de micrófonos (Direction of Arrival DOA).

Por lo tanto, se llega a una manera sencilla de calcular la dirección de la señal de habla en el receptor, empleando para ello los tiempos de llegada de dicha señal a un par de micrófonos.

En caso de trabajar en condiciones de campo cercano, no es posible calcular la DOA de la fuente de voz con respecto al array de micrófonos.

4.2.4. Arrays de micrófonos

Un array de micrófonos consiste en una serie de sensores localizados en posiciones específicas. Se han empleado con frecuencia en tareas de procesamiento de la señal de voz ya que permiten la localización de objetos de forma efectiva además de mejorar la calidad de la señal capturada en comparación con aquella capturada por un único receptor alejado de la fuente [17]. Especialmente interesante es la habilidad de los arrays para filtrar espacialmente, lo que permite el desarrollo de aplicaciones que son capaces de separar la fuente de audio de interés de otras señales no deseadas [18] [36].

Se puede considerar un array de micrófonos como una versión muestreada de un sensor continuo del mismo tamaño que el array, y su respuesta conjunta es modelada como la suma de cada respuesta individual.

Generalmente se considera un array como un conjunto de elementos equiespaciados, y se define su patrón de directividad como sigue [23]:

$$D(f,\theta,\phi) = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f) e^{j\frac{2\pi}{\lambda}\cos\phi\sin\theta nd} = \sum_{n=-\frac{N-1}{2}}^{\frac{N-1}{2}} \omega_n(f) e^{j\frac{2\pi f}{c}\cos\phi\sin\theta nd} \tag{4.7}$$

donde N es el número de elementos, $\omega_n(f)$ es un término denominado peso complejo del elemento n, d la distancia entre los micrófonos y los ángulos θ y ϕ se muestran en la Figura 4.3:

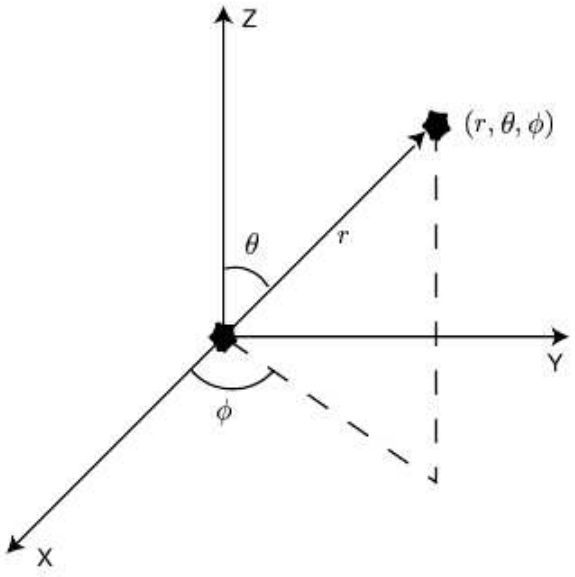


Figura 4.3: Sistema de coordenadas esféricas, siendo r la localización espacial de un punto

Por lo tanto, como se puede ver en la ecuación 4.7 la directividad de un array con indénticos sensores depende de tres factores principales:

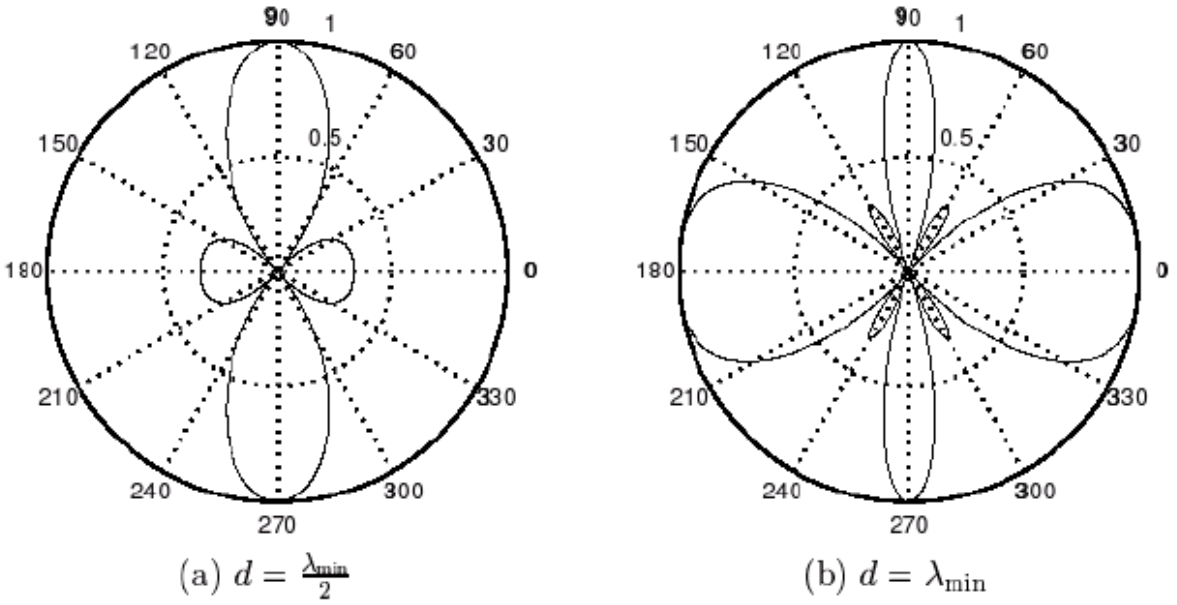
- el número de elementos del array N
- la distancia entre elementos d
- la frecuencia f

Resulta obvio pensar que la mejor opción es disponer de un patrón con elevada directividad (lóbulo principal estrecho), pero en los experimentos llevados a cabo en este trabajo no se puede hacer así, ya que no se conoce a priori las posiciones de los hablantes. Existe por tanto un compromiso entre directividad y resultados en la localización de los hablantes. [37].

4.2.4.1. Aliasing espacial

De acuerdo al teorema de Nyquist, para evitar el aliasing en el muestreo de una señal analógica, la frecuencia de muestreo debe ser mayor de dos veces la máxima frecuencia en la señal muestreada. Esta teoría se puede aplicar también a los arrays de micrófonos equiespaciados [23], obteniendo una relación como la siguiente:

$$d < \frac{\lambda_{min}}{2} \tag{4.8}$$



(a) $d = \frac{\lambda_{min}}{2}$ (b) $d = \lambda_{\min}$

Figura 4.4: Ejemplos de aliasing espacial

donde λ_{min} es la mínima longitud de onda de la señal de interés y d es la distancia entre los micrófonos que debe ser respetada para evitar aliasing espacial, es decir, la aparación de lóbulos de directividad en direcciones no deseadas, como se observa en la Figura 4.4.

4.2.4.2. Conformación de haz o Beamforming

Beamforming es una técnica que permite dirigir el patrón de directividad del array a diferentes direcciones espaciales, Figura 4.5.

Se tiene en cuenta ahora la ecuación 4.7 de la página 39, donde $\omega_n(f)$ es un parámetro que representa el peso complejo aplicado a cada micrófono. Tal como se explica en [38], este conjunto de valores puede ser establecido según diferentes tipos de funciones llamadas ventanas de amplitud. Con ello se consigue controlar la anchura del lóbulo principal y la potencia de los secundarios pertenecientes al patrón de directividad.

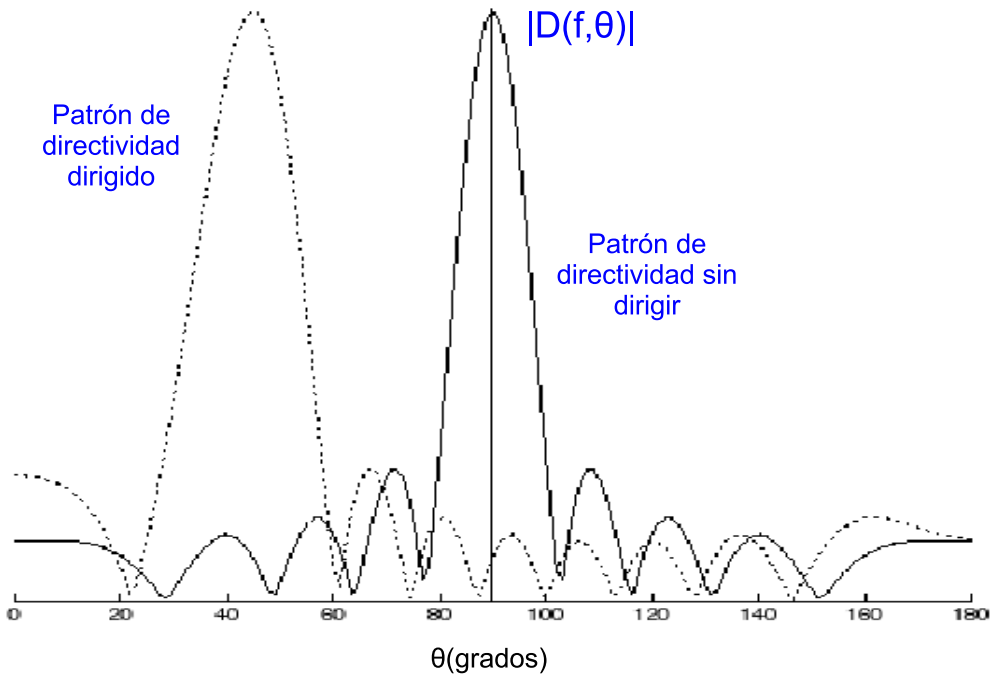


Figura 4.5: Patrones de directividad dirigidos y no dirigidos en la dirección horizontal

A partir de este momento, en todas las figuras se considera que la totalidad de los sensores poseen el mismo peso complejo a la hora de calcular el patrón de directividad:

$$\omega_n(f) = \frac{1}{N} \tag{4.9}$$

El peso se puede expresar de la siguiente manera:

$$\omega_n(f) = a_n(f)e^{j\varphi_n(f)} \tag{4.10}$$

donde $a_n(f)$ y $\varphi_n(f)$ son la amplitud real y la fase del peso respectivamente. Si se modifica la amplitud del parámetro del peso es posibler alterar la forma del patrón de directividad, mientras que si se varía la fase se puede controlar la localización angular del lóbulo principal. Dicha fase se puede representar de la siguiente manera [8]:

$$\varphi_n(f) = -2\pi\frac{\sin\theta'\cos\phi'}{\lambda}nd \tag{4.11}$$

Por lo tanto, el patrón de directividad se dirige a las direcciones θ' y ϕ'. Hay que recalcar que únicamente se está modifcando la fase, por lo que la única diferencia entre el patrón dirigido y el patrón sin dirigir es la dirección a la que apuntan, y no su forma ni niveles.

4.2.5. Algoritmos de localización

Como se ha comentado en la sección 4.2.1 de la página 33, en aquellos casos en que el hablante se encuentra en una habitación cerrada y alejado de los micrófonos, la señal en el receptor tiene

una baja SNR y se encuentra afectada por los efectos de la reverberación. Para disminur estos efectos negativos se emplean con asiduidad los arrays de micrófonos, como se vio en la sección 4.2.4 de la página 39. Por norma general, el hablante no se encuentra en una posición estática, sino que realiza movimientos a lo largo de la habitación; por esto es necesario conocer su posición para así dirigir el patrón de directividad hacia dicha localización.

El factor más importante en todo sistema de localización de hablantes es la precisión, que de acuerdo a [18] depende de cuatro factores fundamentales:

- La cantidad y calidad de los micrófonos
- La posición relativa de los micrófonos con respecto a los otros y a la fuente de voz a analizar.
- Los niveles de ruido y reverberación.
- El número de fuentes activas y su contenido espectral.

Por lo tanto, los resultados del sistema dependen en gran medida del número de micrófonos utilizados en el experimento; a mayor cantidad de micrófonos mejores resultados, especialmente en condiciones acústicas adversas. Se han construido arrays con un gran número de micrófonos (más de 512). Sin embargo, cuando las condiciones acústicas son razonables y la posición de los micrófonos es apropiada, la localización puede ser llevada a cabo con éxito empleando una cantidad menor de micrófonos (por ejemplo 4). En estos casos el comportamiento depende fundamentalmente de la geometría del array y su diseño se relaciona con las condiciones acústicas del entorno y la geometría, así como las condiciones de aplicación específicas.

Además de la exactitud, otras características adoptan una relevada importancia, como es la velocidad o la adaptación a entornos de tiempo real para conseguir una localización efectiva y adaptada a los movimientos.

Con el fin de obtener la localización de la fuente de voz existen diferentes enfoques:

- Aquellos que emplean la diferencia del tiempo de llegada (*Time Difference of Arrival TDOA*).
- Los que usan conceptos de *estimación espectral de alta resolución.*
- Soluciones basadas en maximizar la potencia de respuesta dirigida, *Steered Response Power SRP.*

A continuación únicamente se explica el enfoque TDOA y el SRP, al ser las técnicas empleadas en el desarrollo del algoritmo utilizado en el sistema de experimentación [8].

4.2.5.1. Solución basada en la estimación de de la diferencia del tiempo de llegada TDOA

Esta solución se basa en dos pasos:

- Se caracteriza la diferencia del tiempo de llegada (TDOA) de las señales de voz de los pares de micrófonos separados espacialmente. Existen tres técnicas diferentes:

- La correlación cruzada normalizada (Normalized Cross-Correlation CC)
- El análisis *Crosspower-Spectrum Phase CSP*, también llamado correlación cruzada generalizada (*Generalized Cross-Correlation GCC*)
- Los filtros adaptativos *Least Mean Squared LMS*

Ya se han realizado estudios comparando estas tres técnicas, y llegando a la conclusión de que GCC es el método que mejores estimaciones proporciona.

- Una vez calculada la TDOA, se utiliza la información espacial disponible de los micrófonos, con el fin de generar las curvas hiperbólicas que representan los lugares geométricos donde es probable que el hablante se encuentre.

La principal ventaja de emplear estos métodos para estimar la localización de una fuente de voz es el bajo coste computacional. Sin embargo, los resultados ofrecidos en situaciones con una reducida SNR o efectos reverberantes empeoran considerablemente.

La Correlación Cruzada Normalizada CC Como se ha expuesto anteriormente, una manera de localizar una fuente acústica es conocer el retardo τ, denominado diferencia del tiempo de llegada TDOA entre las señales capturadas por un par de micrófonos separados espacialmente. Las ecuaciones de las señales capturadas por los micrófonos i y j son las siguientes:

$$x_i(t) = \alpha_i \cdot s(t) + n_i(t) \tag{4.12}$$

$$x_j(t) = \alpha_j \cdot s(t + \tau_{ij}) + n_j(t) \tag{4.13}$$

donde *s(t)* es la señal de voz, n_i y n_j corresponden a los ruidos capturados por cada micrófono, α_i y α_j son las atenuaciones y τ_{ij} es el retardo de tiempo entre las señales debido a la distancia que existe entre la fuente y los dos micrófonos.

La forma más utilizada para calcular esta diferencia de tiempo τ_{ij} requiere del cálculo de la Correlación Cruzada (CC) $c_{x_i x_j}(\tau)$, que analiza la similitud entre dos señales diferentes para cada retardo de tiempo τ_{ij}:

$$c_{x_i x_j} = E[x_i(t) \cdot x_j(t - \tau)] \tag{4.14}$$

que se puede expresar de la siguiente manera teniendo en cuenta la ecuación 4.13:

$$c_{x_i x_j}(\tau) = \alpha_i \alpha_j \cdot c_{s_i s_j}(\tau - \tau_{ij}) + c_{n_i n_j}(\tau) \tag{4.15}$$

donde $c_{s_i s_j}(\tau)$ representa la autocorrelación de la señal fuente *s(n)* con el retardo τ.

τ_{ij} se puede calcular maximizando $c_{x_i x_j}(\tau)$ respecto a τ. Sin embargo, debido a que la observación de tiempo es finita, la función sólo puede ser estimada para una ventana temporal de tamaño T. Se le denota a esta estimación $\widehat{c}_{x_i x_j}(\tau)$:

$$\widehat{c}_{x_i x_j}(\tau) = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x_i(t) \cdot x_j(t - \tau) dt \tag{4.16}$$

De esta forma, la estimación del TDOA $\widehat{\tau}_{ij}$ se calcula de la siguiente manera:

$$\widehat{\tau}_{ij} = \arg max_{\tau} \widehat{c}_{x_i x_j}(\tau) \tag{4.17}$$

La Correlación Cruzada Generalizada GCC Existe una versión de la expresión 4.14 denominada Correlación Cruzada Generalizada GCC, que consiste en prefiltrar las señales antes de calcular la correlación para mejorar los resultados obtenidos:

$$c^{(g)}_{x_i x_j}(\tau) = E[(h_i(t) * x_i(t)) \cdot (h_j(t-\tau) * x_j(t-\tau)] \tag{4.18}$$

Otra forma de representar la expresión GCC es la siguiente, donde se expresa en función de los pesos y en términos de frecuencia. Para más detalles [8]:

$$c^{(g)}_{x_i x_j}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_{x_i xj}(\omega) X_i(\omega) X'_j(\omega) e^{-j\omega\tau} d\omega \tag{4.19}$$

Idealmente, con una función apropiada de pesos, la anterior expresión debería presentar un máximo que corresponde exactamente a la TDOA entre los micrófonos i y j. Por lo tanto la TDOA estimada es el instante de tiempo que maximiza $c^{(g)}_{x_i x_j}(\tau)$:

$$\widehat{\tau}_{ij} = \arg max_{\tau} c^{(g)}_{x_i x_j}(\tau) \tag{4.20}$$

Hay que destacar que encontrar $\widehat{\tau}_{ij}$ requiere una búsqueda simple de bajo coste. En general la función $c^{(g)}_{x_i x_j}(\tau)$ proporciona varios máximos. Las amplitudes e instantes de tiempo de los máximos depende de una serie de factores como los niveles de ruido o la reverberación, la distancia entre los micrófonos y la elección de la función de pesos $\Phi_{x_i x_j}(\omega)$.

En la literatura se han propuesto diferentes funciones de filtrado, sin embargo se va a exponer la empleada en este trabajo: la transformada de fase (*The Phase Transform PHAT*).

La técnica PHAT consiste en realizar un filtrado de ruido blanco como estrategia óptima [8]. De esta manera la función de pesado adopta la siguiente expresión:

$$\Phi^{PHAT}_{x_i x_j}(\omega) = \frac{1}{|C_{x_i x_j(\omega)}|} = \frac{1}{|X_i(\omega) X'_j(\omega)|} \tag{4.21}$$

Un inconveniente de esta función de pesos es que el cálculo se lleva a cabo empleando la inversa de los módulos. Además los errores se acentúan en el caso de señales de baja potencia. Para solucionarlo se ha propuesto emplear un filtro paso banda para enfatizar sólo en aquellas frecuencias donde radica más energía.

Como conclusión, el GCC-PHAT es el método que ofrece resultados más interesantes. Es efectivo en situaciones reales y se ha empleado en el desarrollo del algoritmo utilizado en el sistema de experimentación basado en audio utilizado como punto de partida de este trabajo [8].

Implementación de GCC-PHAT Hay varios aspectos a tener en cuenta a la hora de implementar lo descrito en la sección 4.2.5.1 . Como se expuso en la sección 4.2.5.1 de la página 43, no se pueden tomar infinitas muestras de tiempo, sino una ventana temporal para analizar las señales. Además, el sistema no emplea las señales originales y analógicas, sino unas versiones discretas y digitales obtenidas mediante un proceso de muestreo.

Dicho esto, se denotan las señales en tiempo discreto de los micrófonos como $x_1[n]...x_M[n]$. Es necesario dividir en una serie de bloques estas señales y aplicar la transformada discreta de Fourier (DFT) a cada uno de ellos. Cada bloque de datos es normalmente solapado mediante una ventana antes de aplicar la DFT para así mejorar la representación espectral de la señal y eliminar efectos causados por las discontinuidades al final de los bloques. Los bloques de datos consecutivos se solapan en el tiempo para permitir que los datos del final de un bloque estén centrados en el siguiente, dando así el mismo peso a todos ellos. El algoritmo de localización obtiene una estimación para la DFT de cada bloque de datos, asumiendo que la fuente de voz no se mueve en la duración de ese bloque. Como los bloques avanzan en el tiempo, el algoritmo sería capaz de seguir los movimientos de los hablantes. La tasa a la que se proporcionan las estimaciones depende del avance de los bloques, llamado *frame-shift*, y la latencia de cada estimación depende del *frame size*.

La expresión para las señales discretas de los micrófonos $x_1[n]...x_M[n]$ y sus DFTs cuando se segmentan en bloques de longitud N es:

$$x_{m,b}[n] = \omega[n] \cdot x[bA + n] \text{ para n=0...N-1} \tag{4.22}$$

donde $x_{m,b}[n]$ son los datos dentro de la ventana del micrófono m y el bloque b. A es el *frame shift* que define el avance del bloque. El bloque se solapa cuando $A < N$, $\omega[n]$ es la función ventana, usualmente se emplea la ventana Hangin [8].

La DFT del bloque b se puede expresar de la siguiente manera:

$$X_{m,b}[k] = \sum_{n=0}^{N-1} x_{m.b}[n] e^{-jk\frac{2\pi}{K}n} \text{ para k=0...K-1} \tag{4.23}$$

La longitud de la DFT es K y $K > N$. Por lo tanto es necesario añadir ceros al bloque para incrementar su longitud.

La expresión de la función GCC-PHAT basada en la DFT entre los micrófonos i y j del bloque de datos b, $\widehat{c}_{ij,b}$ se puede definir sustituyendo en la ecuación 4.19 de la página 44 la transformada de Fourier de los bloques DFTs previamente definidos:

$$\widehat{c}_{ij,b}(\widehat{\tau}) = \frac{1}{K} \sum_{k=0}^{K-1} \Phi_{ij}[k] X_{i,b}[k] X'_{j,b}[k] e^{jk\frac{2\pi}{K}\widehat{\tau}} = \frac{1}{K} \sum_{k=0}^{K-1} \Phi_{ij}[k] C_{ij,b}[k] e^{jk\frac{2\pi}{K}\widehat{\tau}} \tag{4.24}$$

donde $\Phi_{ij}[k]$ es la versión discreta de la función de pesos $\Phi_{ij}[\omega]$ y $\omega_k = \frac{2\pi k}{K}$ es el índice de frecuencia de la DFT.

Empleando teoremas propios de la DFT [8] se obtiene la siguiente expresión:

$$\widehat{c}_{ij,b}(\widehat{\tau}) = Re[IFFT(\Phi_{ij}[k] X_{i,b}[k] X'_{j,b}[k])](\widehat{\tau}) = Re[IFFT \frac{X_{i,b}[k] X'_{j,b}[k]}{|X_{i,b}[k]||X_{j,b}[k]|}(\widehat{\tau}) \tag{4.25}$$

Esta ecuación muestra cómo implementar la función GCC-PHAT partiendo de la FFT de las señales capturadas por un par de micrófonos. La TDOA estimada entre los dos micrófonos se calcula encontrando el retraso en el cual la función GCC-PHAT $\widehat{c}_{ij,b}(\widehat{\tau})$ es máxima.

La distancia entre los micrófonos d limita físicamente el rango válido para los retrasos temporales. La mayor TDOA puede ser $\frac{d}{c}$, donde c es la velocidad del sonido. Por lo tanto $\tau \in [\frac{-d}{c}, \frac{d}{c}]$. Sin embargo hay un aspecto a mencionar acerca de los valores de τ en el caso de la versión discreta de GCC-PHAT basada en la DFT. La ecuación 4.24 es discreta y todos sus valores son muestreados. Por tanto, existirá una pérdida de precisión al convertir el valor de τ en segundos a su valor discreto $\widehat{\tau}$:

$$\widehat{\tau} = round(\tau[seg] \cdot f_s[seg^{-1}]) \tag{4.26}$$

Asumiendo las condiciones de campo lejano, y tal y como se mostró en la ecuación 4.6 de la página 38, se puede expresar:

$$\widehat{\tau} = round(\frac{d \cdot \sin\theta}{c} \cdot f_s) \tag{4.27}$$

Por lo tanto, la imprecisión depende de la distancia entre los micrófonos *d*, la frecuencia de muestreo *fs*, la dirección de llegada (DOA) y la función de redondeo empleada.

Método de localización basado en TDOA Una vez que se ha estimado la TDOA implementando por ejemplo el algoritmo GCC-PHAT, se pueden emplear los datos junto con la información geométrica para estimar la localización del hablante. Existen varios métodos, pero sólo se muestra uno de ellos: una versión 2D propuesta por Varma en [25]. Asumiendo condiciones de campo lejano, se estima la posición basándose en un método LMS (*Least Mean Square*).

Como se observa en la Figura 4.2 de la página 38, la TDOA, $\widehat{\tau}_{ij}$ entre las señales recibidas por dos micrófonos i y j se expresa de la siguiente manera:

$$d_{ij} \sin\theta = -c\tau_{ij} \tag{4.28}$$

En primer lugar se realiza una estimación de la TDOA a través del método GCC-PHAT para cada posible par de micrófonos del array. Entonces, estas estimaciones se almacenan en un vector τ. Además se guardan las distancias entre todos los posibles pares de micrófonos en otro vector $\mathbf{d}$, que transforma la ecuación 4.28:

$$\mathbf{d} \cdot \sin\theta = -c \cdot \tau \tag{4.29}$$

La ecuación 4.29 representa un sistema de ecuaciones con una única incógnita, la DOA θ, cuyo valor se obtiene mediante el método LMS de la siguiente manera:

$$\widehat{\theta} = \arcsin[(\mathbf{d}^T \cdot d)^{-1} d^T(-c\tau)] \tag{4.30}$$

Es importante recalcar que este método no obtiene la estimación de una posición exacta, sino de una dirección. Sin embargo, esta estimación es útil a la hora de dirigir el patrón de directividad del array a la dirección en la que se localiza el hablante.

4.2.5.2. Solución basada en SRP

Muchas técnicas de procesado de la señal digital se basan en la propiedad que presentan los arrays de micrófonos de focalizar en determinadas posiciones o direcciones del espacio. Estas

técnicas emplean algún tipo de *beamforming* que puede ser utilizado tanto en la señal capturada como en la localización de la fuente. Si se conoce la posición de la fuente, el beamformer puede centrarse directamente en ella para ofrecer una versión mejorada de la señal [37]. En el caso de que la posición no sea conocida, se puede emplear el beamformer para dirigir el array alrededor de un conjunto de localizaciones espaciales en un espacio de búsqueda predefinido. Tras esta fase se emplea un estimador *Maximum Likelihood (ML)* para buscar el máximo pico de potencia en la salida que debe coincidir con la localización del hablante.

El método SRP se ha utilizado con éxito en experimentos con múltiples fuentes, además de suponer un enfoque robusto. Estas son las dos razones por las que se escogió SRP para el desarrollo del algoritmo del sistema empleado en la experimentación [8] y usado como método de localización. La principal desventaja es la alta carga computacional que incrementa progresivamente con el número de micrófonos así como con la cantidad de localizaciones donde buscar una fuente de voz.

El algoritmo SRP-PHAT Como se explicó en la sección 4.2.4.2 de la página 40, las técnicas de *beamforming* permiten dirigir el patrón de directividad del array a diferentes direcciones espaciales. Existe un método denominado *Filter-and-sum beamforming* mediante el cual se le añade une etapa de filtrado a cada micrófono. De este modo, la señal obtenida se expresa de la siguiente manera [2]:

$$y[n] = \sum_{m=0}^{N-1} \sum_{p=0}^{P-1} h_m[p] \cdot x_m[n - p - \tau_m] \tag{4.31}$$

donde $h_m[p]$ es el filtro asociado al micrófono m.

En el dominio de la frecuencia se tiene:

$$Y(\omega, \mathbf{q}) = \sum_{n=1}^{M} W_n(\omega) X_n(\omega) e^{j\omega\Delta_n} \tag{4.32}$$

donde Δ_n es el retardo del micrófono n para dirigir el array a la localización espacial $\mathbf{q}$ y $X_n(\omega)$ y $W_n(\omega)$ son las transformadas de Fourier de la señal del micrófono n y su filtro asociado.

Todo esto es equivalente en el dominio del tiempo a un beamformer que se puede emplear como localizador dirigiendo el array a una zona específica de interés y analizar la potencia de la señal de salida en cada uno de ellos. En el caso de que el array se encuentre dirigido a la dirección de la fuente de habla, la potencia será máxima. La expresión de la potencia para una localización espacial $\mathbf{q}$ se puede expresar como la potencia de salida de un beamformer con filtrado:

$$P(\mathbf{q}) = \int_{-\infty}^{\infty} |Y(\omega_q)|^2 d\omega = \int_{-\infty}^{\infty} Y(\omega_q) Y'(\omega_q) d\omega \tag{4.33}$$

Por lo tanto la localización estimada de $\mathbf{q}$ será:

$$\widehat{q}_s = \arg \max_q P(q) \tag{4.34}$$

Sin embargo, esta potencia puede ser máxima en localizaciones no correspondientes a la fuente de la señal, debido fundamentalmente a ruido o reverberación. Como se vio previamente, la estrategia PHAT de dar un peso a cada componente de frecuencia ha sido probada ofreciendo

resultados satisfactorios. Aunando las ventajas ofrecidas por el *beamforming* junto con las técnicas PHAT se obtiene el algoritmo SRP-PHAT, propuesto por [35] y expresado de la siguiente manera:

$$P(\mathbf{q}) = \sum_{i=1}^{M}\sum_{j=1}^{M}\int_{-\infty}^{\infty}\Phi_{ij}(\omega)X_i\omega X'_j(\omega)e^{j\omega(\Delta_j-\Delta_i)} \tag{4.35}$$

donde:

- $\Phi_{ij}(\omega) = W_i(\omega)W'_j(\omega) = \frac{1}{|X_i(\omega)X'_j(\omega)|} \Leftrightarrow W_n(\omega) = \frac{1}{|X_n(\omega)|}$ son los filtros SRP-PHAT.
- $\tau_{ij} = \Delta_j - \Delta_i$ es la TDOA entre el micrófono i y el j para la onda sonora procedente de la localización $\mathbf{q}$.

SRP en términos de GCC En esta sección se muestra que el SRP de un array de M elementos es equivalente a la suma de las correlaciones cruzadas generalizas GCC de todas las posibles combinaciones de pares de micrófonos. De esta manera, si se incrementa el número de micrófonos lo hará también la robustez del método GCC.

Si se combinan las expresiones 4.35 con la GCC de la señal de dos micrófonos de la ecuación 4.19 de la página 44, una versión en el dominio del tiempo de SRP se puede expresar como la suma de correlaciones cruzadas generalizadas:

$$P(\mathbf{q}) = P(\Delta_1...\Delta_M) = 2\pi\sum_{i=1}^{M}\sum_{j=1}^{M}c_{ij}(\Delta_j - \Delta_i) = 2\pi\sum_{i=1}^{M}\sum_{j=1}^{M}c_{ij}(\tau_{ij}) \tag{4.36}$$

donde $\Delta_1...\Delta_M$ son los retardos en caso de dirigir el array a la localización $\mathbf{q}$ y $c_{ij}(\tau_{ij})$ es el GCC-PHAT de las señales de los micrófonos i y j.

Se calculan las GCC de todos los posibles pares de micrófonos que se encuentran retardadas el tiempo marcado por el retardo para establecer la correcta directividad.

Implementación del SRP-PHAT La ecuación 4.36 define la respuesta dirigida como la suma de funciones GCC. Si se sustituyen las GCCs por su implementación en la ecuación 4.24 de la página 45 se obtiene una potencia estimada de la respuesta dirigida para el bloque b:

$$\widehat{P}_b(\widehat{\Delta}_1\widehat{\Delta}_M) = 2\pi\sum_{i=1}^{M}\sum_{j=1}^{M}\widehat{c}_{ij,b}(\widehat{\tau}_{ij}) = 2\pi\sum_{i=1}^{M}\sum_{j=1}^{M}\frac{1}{K}\sum_{k=0}^{K-1}\Phi_{ij}[k]X_{i,b}[k]X'_{j,b}[k]e^{jk\frac{2\pi}{K}\widehat{\tau}_{ij}} \tag{4.37}$$

De acuerdo a la ecuación 4.25 de la página 45 se puede expresar la fórmula 4.37 en términos de la FFT:

$$\widehat{P}_b(\widehat{\Delta}_1...\widehat{\Delta}_M) = \sum_{i=1}^{M}\sum_{j=1}^{M}Re[IFFT(\frac{X_{i,b}[k]X'_{j,b}[k]}{|X_{i,b}[k]||X_{j,b}[k]|})](\widehat{\tau_{ij}}) \tag{4.38}$$

La ecuación 4.37 muestra una implementación en el dominio del tiempo de SRP denominado *Time SRP (TSRP)* de un bloque capturado por un array de M elementos cuando enfoca a la

localización espacial definida por los retardos de tiempo $\Delta_1 ... \Delta_M$. Sin embargo, aunque estos retardos para establecer la directividad son continuos en la ecuación 4.36 deben ser muestreados en la práctica en la ecuación 4.37. Esto introduce cierta imprecisión en el sistema.

Además, la implementación de SRP en el dominio del tiempo (TSRP) implica una pérdida de precisión al evaluar las funciones GCC muestreadas. Se busca entonces una versión de SRP donde se pueda emplear el tiempo sin tener que discretizarlo. La ecuación 4.35 de la página 48 representa una versión del SRP en el dominio de la frecuencia (FSRP). Con este método la TDOA entre los micrófonos calculada mediante el TSRP es sustituida en el dominio de la frecuencia por multiplicaciones de las funciones GCC por exponenciales complejas evaluadas en los propios retardos de tiempo, con la diferencia de que, esta vez se puede hacer uso del tiempo en unidades reales sin tener que discretizar.

4.2.5.3. Estrategias adicionales

En el desarrollo del algoritmo SRP-PHAT [8] se han empleado las técnicas descritas anteriormente. Además, y para obtener mejores resultados y soluciones más óptimas se añadieron algunas mejoreas que se describen brevemente a continuación.

Búsqueda coarse to fine Como se explicó en la introducción de SRP en la sección 4.2.5.2 de la página 46, el principal problema que presenta este algoritmo es la alta carga computacional.

Para solventar esto, se hace uso de la relación espacio-frecuencia del sonido. A las frecuencias altas le corresponden pequeñas longitudes de onda que pueden explorar el espacio de una forma más precisa, mientras que a las frecuencias más bajas le corresponden longitudes de onda mayores que integran grandes áreas del espacio. Zotkin y Duraiswami [39] llevaron a cabo experimentos para obtener una relación entre la anchura del pico de energía espacial y la frecuencia de la fuente. Los resultados ofrecieron la siguiente relación:

$$b \simeq \frac{2\lambda}{5} \text{ donde b es la anchura del pico del beamformer} \tag{4.39}$$

Se deduce que empleando las bajas frecuencias se puede explorar la enegría procedente de amplias áreas espaciales, ya que la anchura del pico es más grande. Así se puede analizar el espacio sólo evaluando unos pocos y distantes puntos del espacio. Igualmente si se eleva la frecuencia que se emplea en las tareas de localización se pueden evaluar regiones cada vez más pequeñas.

Para emplear este método, en primer lugar se selecciona una frecuencia de corte baja que define un reducido número de áreas amplias. Una vez seleccionadas aquellas que contienen más cantidad de energía se divide en partes equiespaciadas y se exploran una a una con una frecuencia de corte el doble al anterior caso. Se procede así tantas veces como sea deseado para obtener la precisión buscada.

Consideraciones de la relación señal ruido Es recomendable descartar los bloques de voz en los que existen pausas o potencias de voz muy bajas, ya que los resultados son estimaciones de localizaciones pobres. Para evitar este efecto se establece un límite de potencia, calculado teniendo en cuenta los primeros instantes de tiempo donde se supone que no existe actividad de voz. Las muestras de la señal que se encuentren por debajo de ese límite son descartadas en los cálculos de la localización. Se pueden establecer límites fijos o adaptativos [8].

Estimación de la confianza en la localización Se desea tener información a priori de la fiabilidad de las estimaciones que el algoritmo va a ofrecer. De esta manera se puede realizar un diseño donde se dé más énfasis a aquellos factores que ofrecerán mejores resultados, intentando disminuir los que empeoran la localización.

Existen multitud de parámetros iniciales que determinan los resultados finales: la geometría de los micrófonos, la componente espectral de la señal, la relación señal ruido, etc. Por lo tanto es necesario medir todos estos parámetros para poder minimizar el efecto de aquellos que empeoran los resultados obtenidos. Para ello se emplean funciones de pesado o redes neuronales [8].

Técnicas de interpolación Cuando se trabaja con señales digitales, se asume que el retardo de tiempo entre ellas es un número finito de muestras. Pero esto no es siempre así, por lo que se comete cierta imprecisión, imprecisión que depende de múltiples factores. En la ecuación 4.26 de la página 46 se comprueba que a mayor frecuencia de muestreo la precisión será mayor al evaluar las funciones GCC.

Sin embargo la frecuencia de muestreo en una grabación es fija y no hay posibilidad de variarla. Se ha desarrollado una técnica [25] que interpola las funciones discretas GCC-PHAT llevando a cabo un *zero-padding* en el dominio de la frecuencia. Hay que destacar que el *zero-padding* en el dominio de la frecuencia es equivalente a interpolar en el dominio del tiempo.

Técnicas de filtrado Como se comentó anteriormente, es interesante comprobar el funcionamiento del algoritmo en distintas bandas de frecuencia con el fin de testear el efecto de las diferentes componentes en las estimaciones finales. Con esta intención se han aplicado diferentes filtros paso bajo, paso o alto o paso banda.

Estas técnicas fueron empleadas por [35], que tuvo en cuenta las bandas de energía en las que se concentra la mayor parte de ella en el caso de señales sonoras de habla humana. Esta energía se encuentra principalmente en frecuencias bajas; de esta forma en las frecuencias altas más aptas para realizar búsquedas de forma más precisa en el espacio por su menor longitud de onda, no existe gran cantidad de energía del habla y por tanto se da una baja SNR.

Uso de información geométrica Como se explicó con anterioridad, cualquier información que se pueda conocer a priori sobre el comportamiento del algoritmo es fundamental para mejorar las localizaciones ofrecidas. Por lo tanto, emplear el conocimiento que se tiene sobre las características físicas del entorno en el que se encuentre el hablante resulta de gran utilidad, por ejemplo la geometría, límites y áreas muertas o áreas donde no se va a encontrar la fuente de voz.

4.3. Localización basada en información procedente de visión

4.3.1. Introducción

En este trabajo se ha empleado un sistema de experimentación basado en una arquitectura cliente servidor, explicado con más profundidad en el Manual de Usuario de la página 139. Este sistema compone un espacio de ocupación partiendo de unas imágenes de entrada procedentes de distintas cámaras y le aplica un algoritmo de estimación de posiciones para obtener la localización de los objetos presentes en la sala.

En esta sección se explica de forma breve el estado del arte de la localización basada en visión, aquellos conceptos teóricos involucrados en la formación la imagen a partir de una cámara, cómo se obtiene su proyección en el plano deseado a través de la aplicación de la matriz de homografía y cómo se genera un grid partiendo de varias imágenes. Por último se detalla el sistema de estimación de la posición aplicado, el Filtro de Partículas.

4.3.2. Estado del arte en la detección, localización y estimación de pose con sensores de vídeo

La visión constituye uno de los mecanismos sensoriales de percepción más importantes en el ser humano, en cuanto a la cantidad de información que se puede obtener a partir del mismo. La utilización en conjunto de cámaras de visión y computadoras de gran capacidad intentan simular este sentido humano para así emplear la gran cantidad de información dentro del ámbito de aplicación.

En el caso del uso de la información visual, es imprescindible en primer lugar realizar un modelado de los *targets*. Para ello se identifica un vector de estados y unos modelos de actuación y observación. En el vector de estados se identifican las variables estimadas a la salida del algoritmo de seguimiento, implementado a partir de las características de observación definidas en el modelo de observación y con una evolución temporal caracterizada por el modelo de actuación. La correcta elección de dichos modelos supondrá la obtención de unos buenos resultados. A este respecto actualmente la comunidad científica se encuentra dividida en dos tendencias:

- Modelos complejos y científicos: centrados en el aprovechamiento de las modernas técnicas de procesado de imagen para obtener un modelo específico y robusto de los *targets* a seguir, basados en el color, la forma, el contorno, etc. Como contrapartida en muchos casos se simplifica el algoritmo de estimación de la posición. Este tipo de modelos se emplean en trabajos como [40] y [41].

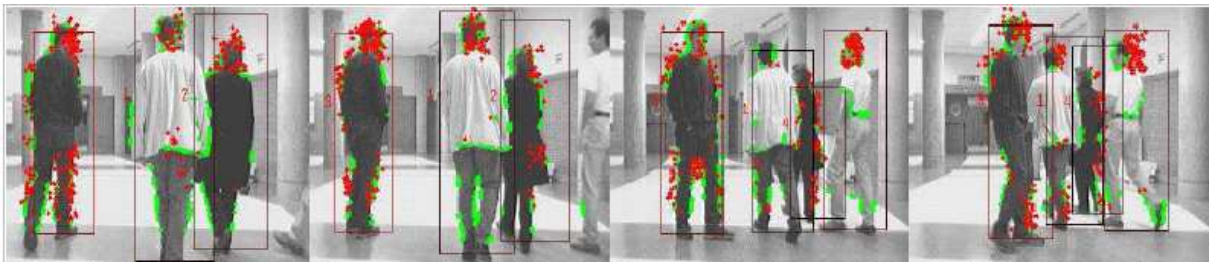


Figura 4.6: Resultados extraídos de los experimentos de [Marrón, 2008]

- Modelos sencillos y generales: Estos modelos focalizan su esfuerzo en la estimación de la posición y seguimientos empleando modelos sencillos. Esto redunda en una generalidad de los *targets* y una reducción de los tiempos de cómputo, ayudando a conseguir especificaciones de tiempo real. Este tipo de modelos se proponen en [42] y [10] (Figura 4.6).

El principal problema existente en tareas de localización y seguimiento basado en vídeo es el seguimiento de múltiples *targets*. El motivo principal de esta complejidad son las diferentes características del entorno. Se divide el problema en dos bien diferenciados [43]:

1. Estimación del vector de estados de cada *target*.

2. Asociación del vector de estados con alguno de los *targets* presentes en el entorno.

Estos dos procesos se pueden llevar a cabo mediante algoritmos determinísticos, probabilísticos o una combinación de ambos [10].

Las soluciones más eficaces para los algoritmos de estimación son las probabilísticas basadas en el teorema de Bayes [44], y las distintas particularizaciones de esta teoría dan lugar a distintos algoritmos [45].

En cuanto a los algoritmos de asociación son de nuevo los probabilísticos los que mejores prestaciones ofrecen, fundamentalmente los basados en PDA (*Probabilistic Data Association*) y su combinación con el filtro de partículas (PDAF [46]), o los basados en JPDA (*Joint Probabilistic Data Association*) y su combinación con el filtrado de partículas (JPDAF [47]).

Figura 4.7: Resultados en la estimación de pose obtenidos en [Lanz, 2006]

En cuanto a la estimación de pose se refiere, existe una cantidad de trabajos superior que para el caso de audio. En este punto destacan los trabajos realizados en ITC [48] (Figura 4.7) y el instituto suizo IDIAP [49] [50] [51] [52] y [53]. Todos ellos hacen uso intensivo de estrategias basadas en filtrado de partículas y modelos de observación y actuación específicos.

4.3.3. Formación de la imagen procedente de un sistema de visión

Una cámara convierte una imagen en tres dimensiones en otra en dos dimensiones, por lo que se realizan una serie de transformaciones de la imagen origen para la conversión de un espacio a otro. La conversión entre los dos espacios se explica mediante el modelo de la cámara, siendo en este caso el modelo pinhole.

4.3.3.1. Idea del modelo pinhole

Una idea intuituva del modelo pinhole consiste en hacer un agujero infinitamente estrecho sobre un material infinitamente delgado, tras el que se coloca una superfcie sensible a la luz, siendo en ella donde se genera la imagen [54]. Esta idea se observa en la Figura 4.8.

De esta manera se comprende de forma sencilla cómo se forma una imagen en una cámara pinhole. No obstante, para el posterior estudio se considera que la imagen se genera delante del punto de proyección de los rayos, quedando así la imagen no invertida, Figura 4.9.

4.3.3.2. Modelo de la cámara pinhole

El modelo de la cámara pinhole se considera como la proyección de un punto del espacio 3D sobre un plano [55], donde el centro de la proyección es el centro óptico de la cámara, *C*.

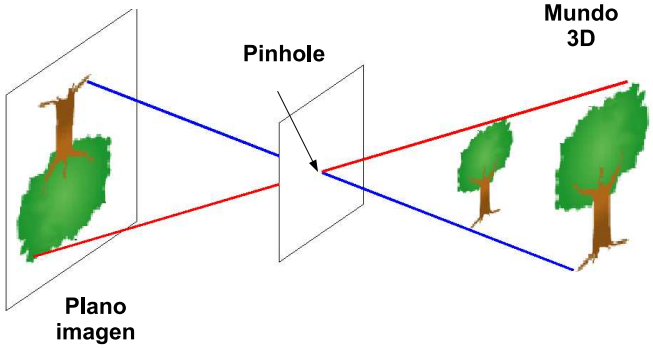


Figura 4.8: Idea intuitiva del modelo de la cámara pinhole

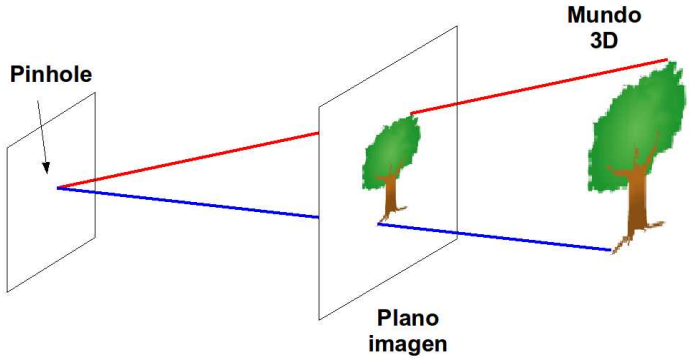


Figura 4.9: Imagen no invertida de la cámara pinhole

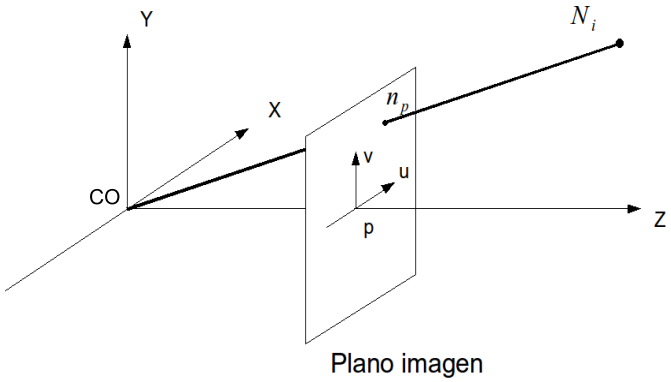


Figura 4.10: Geometría de formación de la imagen en una cámara pinhole

El plano se encuentra a una distancia f de dicho centro, conocido como plano imagen. De esta manera, un punto cuyas coordenadas son $N_i = (X_i, Y_i, Z_i)^T$ se proyecta en el plano imagen en el cruce de la recta $\overline{N_i C}$ y el plano imagen. A dicho punto se le conoce como n_p, Figura 4.10

Donde:

- $CO \rightarrow$ Centro óptico de proyección central. Es el centro de la cámara y representa el origen de coordenadas de ésta. Por él pasan todos los rayos de luz procedentes del espacio que generan la imagen en el sensor.
- $Z \rightarrow$ Eje o rayo principal. Línea perpendicular al plano imagen que pasa a su vez por el centro óptico.
- $p \rightarrow$ Punto principal que coincide con la intersección entre el plano imagen y el eje principal, siendo el centro del plano imagen.
- $N_i \rightarrow$ Representa un punto cualquiera en el espacio cuyo origen es C y que se proyecta sobre el plano imagen.
- $n_p \rightarrow$ Punto del espacio proyectado sobre el plano imagen.

En este modelo es posible realizar un cambio de origen y escala en el plano imagen, así como una rotación y traslación del sistema de referencia, obteniendo finalmente la siguiente expresión:

$$\widetilde{n}_k = K[R|t]\widetilde{N}_w \tag{4.40}$$

Donde:
$K \rightarrow$ es la matriz de calibración de la cámara.
$R \rightarrow$ es la matriz de rotación.
$t \rightarrow$ es la matriz de traslación.
$\widetilde{N}_w \rightarrow$ es un punto del espacio referido a un sistema de coordenadas elegido, en coordenadas homogéneas.
$\widetilde{n}_k \rightarrow$ es un punto en el plano imagen con origen en un extremo del mismo, resultado de la proyección de N_w sobre el centro óptico, en coordenadas homogéneas.

4.3.4. Cálculo de la matriz de homografía

La matriz de homografía H relaciona los puntos en un plano π con los puntos pertenecientes al plano imagen, dado un modelo de cámara pinhole con sus correspondientes matrices *K, R, t* y dado un plano en el espacio tridimensional π; ver Figura 4.11.

Para generar la matriz de homografía se sitúa el sistema de cooordenadas de tal manera que el eje Z_w sea perpendicular al plano.

La expresión de la matriz de homografía para una altura determinada z_0 es la siguiente:

$$H = \begin{bmatrix} C_{11} & C_{12} & K_1 t + z_0 * C_{13} \\ C_{21} & C_{22} & K_2 t + z_0 * C_{23} \\ C_{31} & C_{32} & K_3 t + z_0 * C_{33} \end{bmatrix} \tag{4.41}$$

Donde:

- $C \rightarrow$ Matriz resultado de multiplicar la matriz de calibración de la cámara K y la matriz de rotación R.

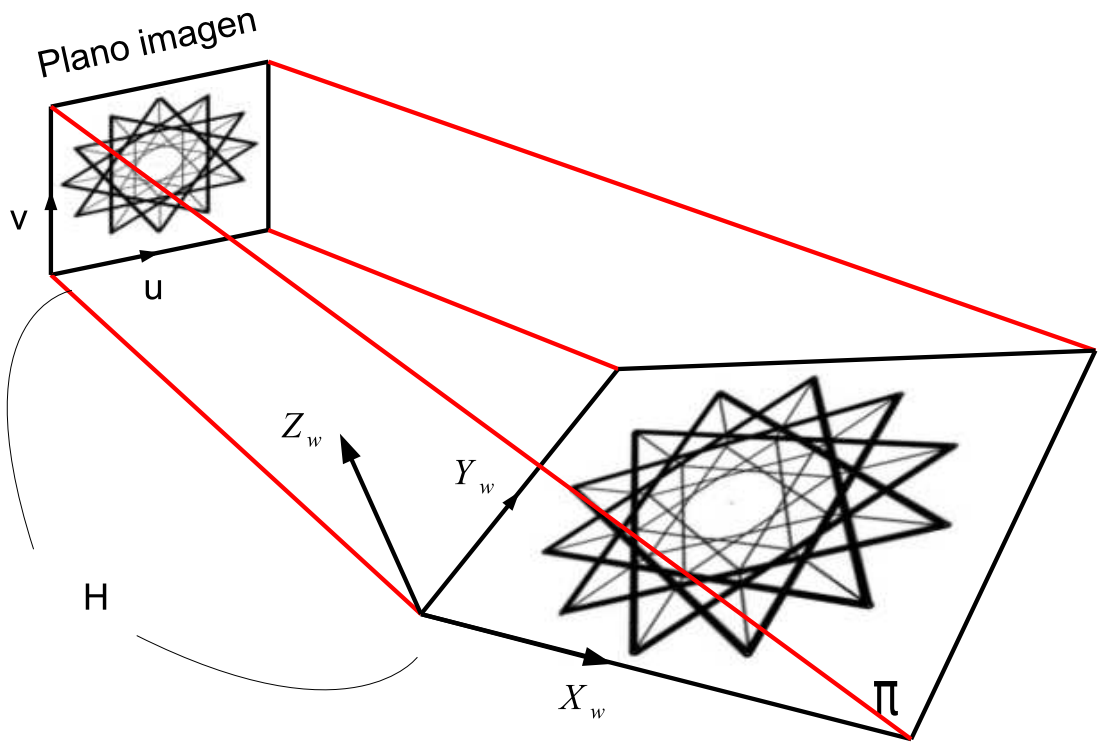


Figura 4.11: Relación de la matriz de homografia

- $K_i \rightarrow$ Fila i-ésima de la matriz de calibración.

De esta manera, un punto en una plano dado, N_w, se proyecta en la imagen en el punto n_k de acuerdo a la siguiente expresión:

$$\widetilde{n}_k = H\overline{N}_w \tag{4.42}$$

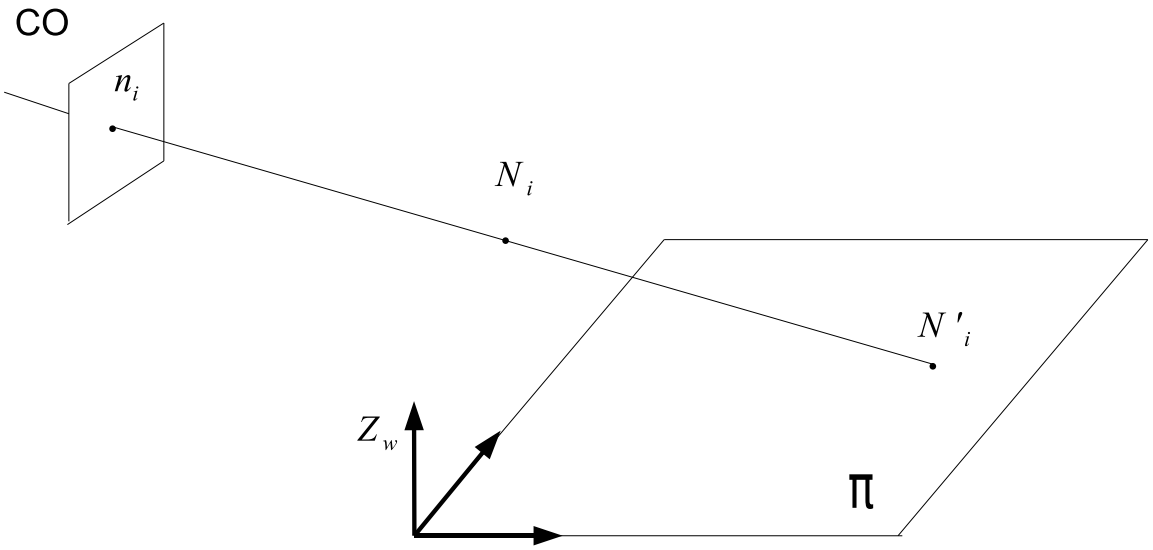


Figura 4.12: Proyección de un punto no perteneciente al plano sobre el plano imagen

En el caso de que existan puntos no contenidos en el plano π, se puede realizar su proyección sobre dicho plano. Se tiene un punto N_i fuera del plano π y una matriz de homografía H que relaciona los puntos del plano π y el plano imagen. Con el punto N_i y el centro óptico de la cámara C se crea una recta, donde todos los puntos de dicha recta se proyectan sobre el mismo punto en el plano imagen n_i. Esta explicación se puede ver en la Figura 4.12.

Si se aplica la matriz homografía inversa H^{-1} al punto n_i se obtiene el punto en el plano π que genera n_i, es decir, $N'i$.

4.3.5. Generación de un grid de ocupación partiendo de múltiples cámaras

Gracias al uso de la matriz de homografía es posible realizar diversas aplicaciones, como por ejemplo la generación de un grid de ocupación [54]. Un grid es la división en forma cuadriculada de un plano, donde la resolución de dicho plano se adapta a la nueva subdivisión. La ocupación en un plano se refiere a aquellas áreas del mismo en la que se encuentran situados objetos. Por lo tanto, el grid de ocupación representa la ocupación en un plano subdividido en un grid, resultando una imagen con dos posibles valores: ocupado o no ocupado.

Para formar el grid hay que apoyarse en lo comentado en la sección Cálculo de la matriz de homografía de la página 54, donde se explica que un punto fuera del plano π se proyecta a través

de la homografía inversa H^{-1} en un punto del plano π; dicho punto depende de la colocación de la cámara.

De esta manera, si se aplica H^{-1} a las M imágenes de las M cámaras, se obtienen las M proyecciones sobre el plano π.

Una vez que se han hecho las M transformaciones de las imágenes, el grid de ocupación en el plano π se compone por la superposición de las M transformaciones, siendo las áreas ocupadas aquellas en las que las M proyecciones se superponen.

En la Figura 3.13 se muestra un ejemplo donde se puede ver una misma escena capturada por dos cámaras diferentes, la segmentación de fondo de cada una de ellas (resta del fondo a la imagen) y sus respectivas proyecciones tras aplicar la homografía inversa (a una altura de 1,70 m). En la figura 4.14 se observa el grid resultante de la superposición de ambas proyecciones:

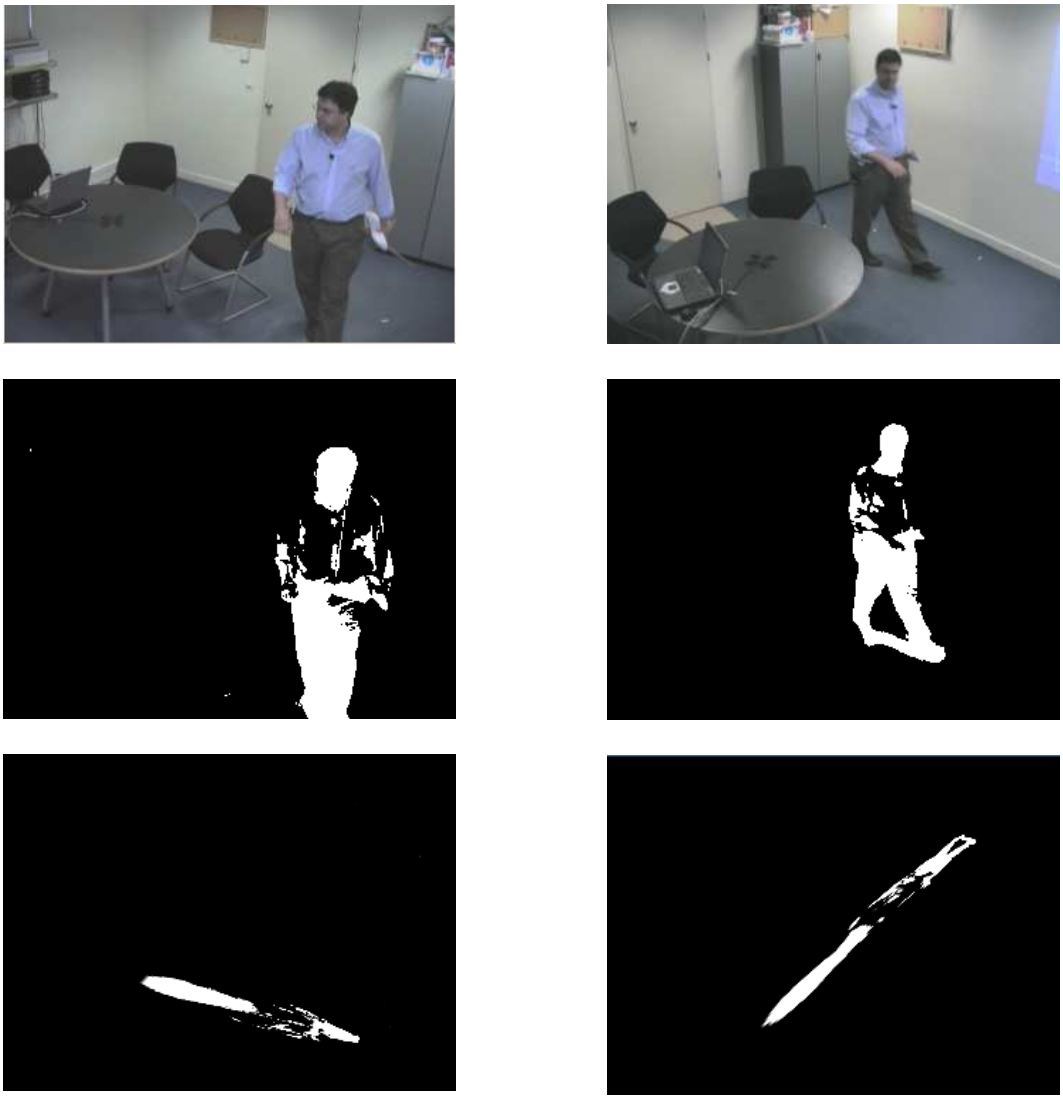


Figura 4.13: Imágenes de las cámaras (arriba), segmentación de fondo (medio), proyección de las imágenes (abajo)

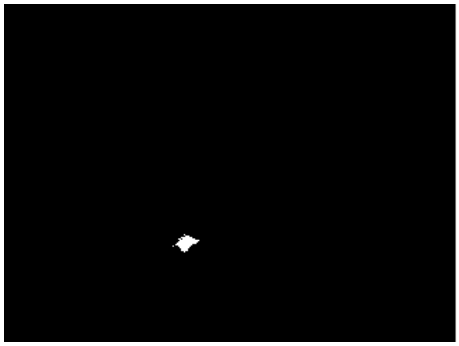


Figura 4.14: Resultado de la superposición de las dos proyecciones

4.3.6. Estimación de la posición basada en información procedente de visión

Una vez que se dispone del grid de la imagen, es posible llevar a cabo un proceso de estimación de la posición con el fin de conseguir un seguimiento adecuado. Para ello se emplea el algoritmo "Filtro de Partículas Extendido con proceso de Clasificación" (*Extended Particle Filter with a Clustering Process XPFCP*) [10].

El algoritmo XPFCP está compuesto por un Filtro de Partículas extendido que permite realizar la estimación del vector de estados $\overrightarrow{x}_t$, y de un proceso de clasificación que hace las veces de proceso de asociación y aumenta la robustez del filtro de partículas comentado [56].

4.3.6.1. El estimador Filtro de Partículas Extendido

Los Filtros de Partículas (PF) son algoritmos de estimación que permiten modelar el comportamiendo del sistema de interés mediante funciones de probabilidad multimodales, por lo que son especialmente adecuados en aplicaciones de estimación para sistemas cuyos modelos son no lineales y tienen asociados ruidos que pueden ser no Gaussianos. El Filtro de Partículas es un método de Monte Carlo recursivo.

En primer lugar se muestran una serie de definiciones necesarias para la comprensión de la explicación:

- Se denominan medidas $\overrightarrow{y}$ a los datos obtenidos directamente de la imagen.
- Se denomina vector de estados al vector $\overrightarrow{x}_t = (x_t\ y_t\ z_t\ vx_t\ vy_t)$, donde x_t, y_t, z_t corresponden a la posición de la partícula en las tres coordenadas, y vx_t, vy_t la velocidad de la partícula en la coordenada x e y.
- Se denominan partículas al par $(\overrightarrow{x}_t^{(i)}, \widetilde{w}_t^{(i)})$, siendo $\overrightarrow{x}_t^{(i)}$ el vector de estado de la partícula i-ésima en el instante de tiempo t, y $\widetilde{w}_t^{(i)}$ el peso asociado a la partícula i-ésima en el instante t.
- Se denomina $p(\overrightarrow{x}_t|\overrightarrow{y}_{1:t})$ a la función de densidad de probabilidad de que un objeto dado se encuentre en el estado $\overrightarrow{x}_t$ teniendo en cuenta las medidas desde el instante de tiempo 1 a t.

La idea principal de un PF es representar y actualizar la función de probabilidad *a posteriori* $p(\overrightarrow{x}_t|\overrightarrow{y}_{1:t})$ mediante un conjunto de muestras aleatorias $S_{t-1} = \{\overrightarrow{x}_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{n}$ que tienen pesos asociados y calcular el estado estimado $\overrightarrow{x}_{t+1|t}$ a través de esas muestras y esos pesos [57].

Una descripción detallada de la base matemática del PF se puede encontrar en [58] y [59]. En el modelo dinámico del Filtro de Partículas, el vector de estado inluye la posición y la velocidad en coordenadas Cartesianas. Dicho modelo se muestra a continuación:

$$\overrightarrow{x}_{t+1|t} = A \cdot \overrightarrow{x}_t + \overrightarrow{q}_t = \begin{bmatrix} 1 & 0 & 0 & T_s & 0 \\ 0 & 1 & 0 & 0 & T_s \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} + \overrightarrow{q}_t \tag{4.43}$$

$$\overrightarrow{y}_t = H \cdot \overrightarrow{x}_t + \overrightarrow{r}_t = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ y_t \\ z_t \\ vx_t \\ vz_t \end{bmatrix} + r \tag{4.44}$$

donde $\overrightarrow{x}_t$ es el vector de estado, $\overrightarrow{y}_t$ el vector de medidas, $\overrightarrow{q}_t$ y $\overrightarrow{r}_t$ indican el ruido del modelo y de las medidas respectivamente.

El bucle principal del PF estándar, llamado SIR (Sequential Important Resampling) [59] [60] comienza en el instante t con un set $S_{t-1} = \{\overrightarrow{x}^{(i)}_{t-1}, \widetilde{w}^{(i)}_{t-1}\}^n_{i=1}$ de n muestras o partículas aleatorias, reprensentando la probabilidad del vector de estado $p(\overrightarrow{x}_{t-1}|\overrightarrow{y}_{1:t-1})$, estimado en el instante *t-1*. El resto del proceso se desarrolla a lo largo de tres pasos:

1. **Predicción**: Las partículas se propagan a través del modelo de estado del sistema bajo estudio para obtener un nuevo set $S_{t|t-1} = \{\overrightarrow{x}^{(i)}_{t|t-1}, \widetilde{w}^{(i)}_{t-1}\}^n_{i=1}$ que representa la distribución *a priori* del vector de estado en el instante t, $p(\overrightarrow{x}_t|\overrightarrow{y}_{1:t-1})$.

2. **Corrección**: El peso de cada partícula $\overrightarrow{w}_t = \{w^{(i)}_t\}^n_{i=1} \equiv (\overrightarrow{x}_{0:t})$ se obtiene comparando el vector de salida $\overrightarrow{y}_t$ y el valor predicho basado en la estimación $h(\overrightarrow{x}_{t|t-1})$, lo que implica que se obtiene directamente de la función de verosimilitud $p(\overrightarrow{y}_t|\overrightarrow{x}_t)$ del modelo elegido:

$$w(\overrightarrow{x}_{0:t}) = w(\overrightarrow{x}_{0:t-1}) \cdot \frac{p(\overrightarrow{y}_t|\overrightarrow{x}_t) \cdot p(\overrightarrow{x}_t|\overrightarrow{x}_{t-1})}{q(\overrightarrow{x}_t|\overrightarrow{x}_{0:t-1}, \overrightarrow{y}_{1:t})} \longrightarrow w(\overrightarrow{x}_{0:t}) = w(\overrightarrow{x}_{0:t-1}) \cdot p(\overrightarrow{y}_t|\overrightarrow{x}_t) \tag{4.45}$$

3. **Selección**: Usando el vector de pesos $\overrightarrow{w}_t = \{w^{(i)}_t\}^n_{i=1}$ calculado en el paso anterior y aplicando el método de muestreo elegido, un nuevo set $S_t = \{\overrightarrow{x}^{(i)}_t, \widetilde{w}^{(i)}_t\}^n_{i=1}$ es obtenido con las partículas más probables, que representarán la nueva creencia $p(\overrightarrow{x}_t|\overrightarrow{y}_{1:t})$ o probabilidad *a posteriori* del vector de estado. Este conjunto de muestras es usado como entrada para la iteración del algoritmo en el instante *t+1*.

En la aplicación, el Filtro de Partículas estándar se puede usar para realizar la estimación de un único objeto definido a través de su modelo de movimiento, pero no para el caso en que aparezcan objetos ya que en este caso no se asociará ninguna partícula al nuevo objeto a no ser que su vector de estados fuese similar al del ya estimado.

Para adaptar este método a un número variable de objetos se introducen algunas modificaciones:

- Un nuevo paso de reinicialización: En este paso previo a la predicción, n_m de las n partículas totales son sustituidas por medidas procedentes del vector de observación $\overrightarrow{y}_{t-1}$. Es la única

forma de obtener una creencia que represente a múltiples hipótesis sin que sean rechazadas en el paso de selección.

- En el paso de corrección: Por una parte, sólo $n - n_m$ muestras del set de partículas son extraídas en este paso, y por otra el cálculo de los pesos de las partículas cambia al emplear para su cómputo el proceso de clasificación del que se habla posteriormente.

4.3.6.2. El proceso clasificador

El clasificador aumenta la robustez del proceso de estimación probabilístico y le permite adaptarse al seguimiento de un número variable de objetos.

El algoritmo de clasificación organiza el set de medidas en grupos o clases que representan todos los objetos que se encuentran en una escena [61].

Se ha usado como algoritmo clasificador en el XPFCP una versión adaptada del K-Medias para un número variable de objetos. Este método de clasificación es recursivo y booleano y se caracteriza por los siguientes aspectos [62]:

- Los elementos pertenecientes a cada grupo deben ser similares entre sí, y diferentes a los de los otros grupos en relación a la característica que se emplee como variable de clasificación.
- Los datos clasificados han de pertenecer a un grupo y sólo a uno.
- No puede quedar ninguna clase vacía.

El algoritmo K-Medias consta de las siguiente etapas:

1. Se eligen k datos del set Y para utilizarlos como valor inicial de los centroides de las clases a crear.
2. Se calcula la distancia de cada dato a todos los centroides y se asocia el dato a la clase más próxima.
3. Realizada la asignación de todo el set de datos Y, se recalcula el valor del centroide de cada clase, como la media geométrica del conjunto de datos asignados a la clase en el espacio de definición de la distancia Euclídea.
4. Se analiza si ha habido algún cambio en la última iteración en el valor del centroide de cada clase o, lo que es equivalente, en los miembros asignados a cada clase. Si es así, el algoritmo se repite desde el paso 2.
5. En caso contrario, o si el número de veces que se ha ejecutado el proceso excede a un límite predefinido, el algoritmo finaliza, antes de lo cual se eliminan las clases que han quedado sin miembros asociados.

Uno de los principales inconvenientes de este segmentador es que los ruidos reciben el mismo tratamiento que el resto de datos, y no se pueden distinguir ni asociarse a algún grupo especial. Para dar solución a esto, se incluye un proceso de validación que aprovecha la aparición esporádica de dichos ruidos para conseguir filtrarlos.

Además, se incluye un proceso de estimación inicial de los centroides de las clases con el fin de mejorar la convergencia y el tiempo de ejecución del segmentador.

En la Figura 4.15 se muestra un flujograma general del XPFCP que se emplea en el sistema de experimentación.

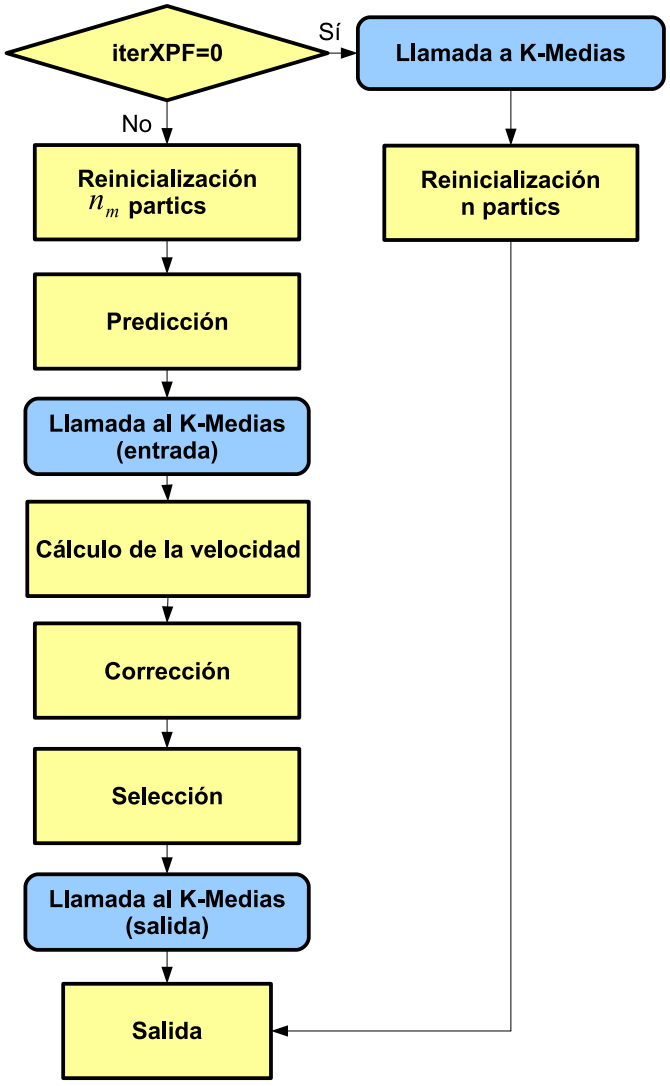


Figura 4.15: Flujograma del algoritmo XPFXP

4.4. Localización basada en fusión audiovisual

En los últimos tiempos, la fusión multisensorial es un campo de investigación en auge, ya que cada uno de los sensores tiene sus propias debilidades y fortalezas, compensándose los unos a los otros. Esta fusión tiene como principal objetivo aumentar la robustez y fiabilidad del algoritmo final.

La información visual es una fuente no afectada por el entorno acústico, por lo que se ha demostrado que su incorporación a los sistemas de reconocimiento automático del habla puede elevar sus prestaciones [63] [64]. Existen tres razones que justifican la utilización de información visual como ayuda para el reconocimiento del habla [65]:

- ayuda a localizar al locutor
- contiene información segmental de habla que complementa al vídeo
- ayuda a resolver ambigüedades con información adicional

Como se ha comentado, la comunidad investigadora ha destacado en los últimos años la importancia de emplear técnicas de fusión multisensorial, a pesar de lo cual, las soluciones existentes basadas únicamente en técnicas de localización mediante métodos acústicos o visuales son más abundantes que aquellas basadas en fusión audiovisual.

Las principales aplicaciones de la fusión audiovisual son en la actualidad reuniones, vigilancia, teleconferencias, "Espacios Inteligentes" y los interfaces persona-máquina. Estas aplicaciones son semejantes a las mostradas en los apartados de audio o visión.

Una posible clasificación de los métodos que emplean fusión audiovisual se puede realizar basándose en las metas perseguidas por cada uno (*tracking de un único o múltiples targets*), la configuración de los sensores audiovisuales y el entorno específico en el que se desarrolla. Sin embargo, existe una clasificación más importante desde el punto de vista del diseño arquitectural que se presenta a continuación y que se observa de forma gráfica en la figura 4.16:

- Orientados a Sistema (*System Oriented*): Estos algoritmos se basan en explotar las características que ofrecen los algoritmos de *tracking* de audio y visión de manera independiente, y a partir de la estimación generada por cada uno de ellos llegar a la solución óptima de combinación de dicha información.
- Orientados a Modelo (*Model Oriented*): Estos algoritmos tratan de obtener una formulación matemática conjunta que saque partido a las fortalezas de cada tipo de sensor en las diferentes etapas del algoritmo, de manera que se genera directamente una estimación conjunta.

En la Figura 4.16 se observa el esquema con los dos tipos de algoritmos, donde la información de cada tipo de sensor ($y_a(t)$ en el caso de audio e $y_v(t)$ para el sensor de vídeo) se entrega bien a algoritmos de *tracking* independientes para audio y visión o bien a un algoritmo conjunto de fusión audiovisual.

La mayor parte del trabajo realizado hasta el día de hoy se centra en un único locutor [16], ya sea en un escenario con una única persona presente [66] [67]; o un escenario con múltiples hablantes donde sólo existe un único locutor activo [68] [69] (Figura 4.17). En todos estos casos

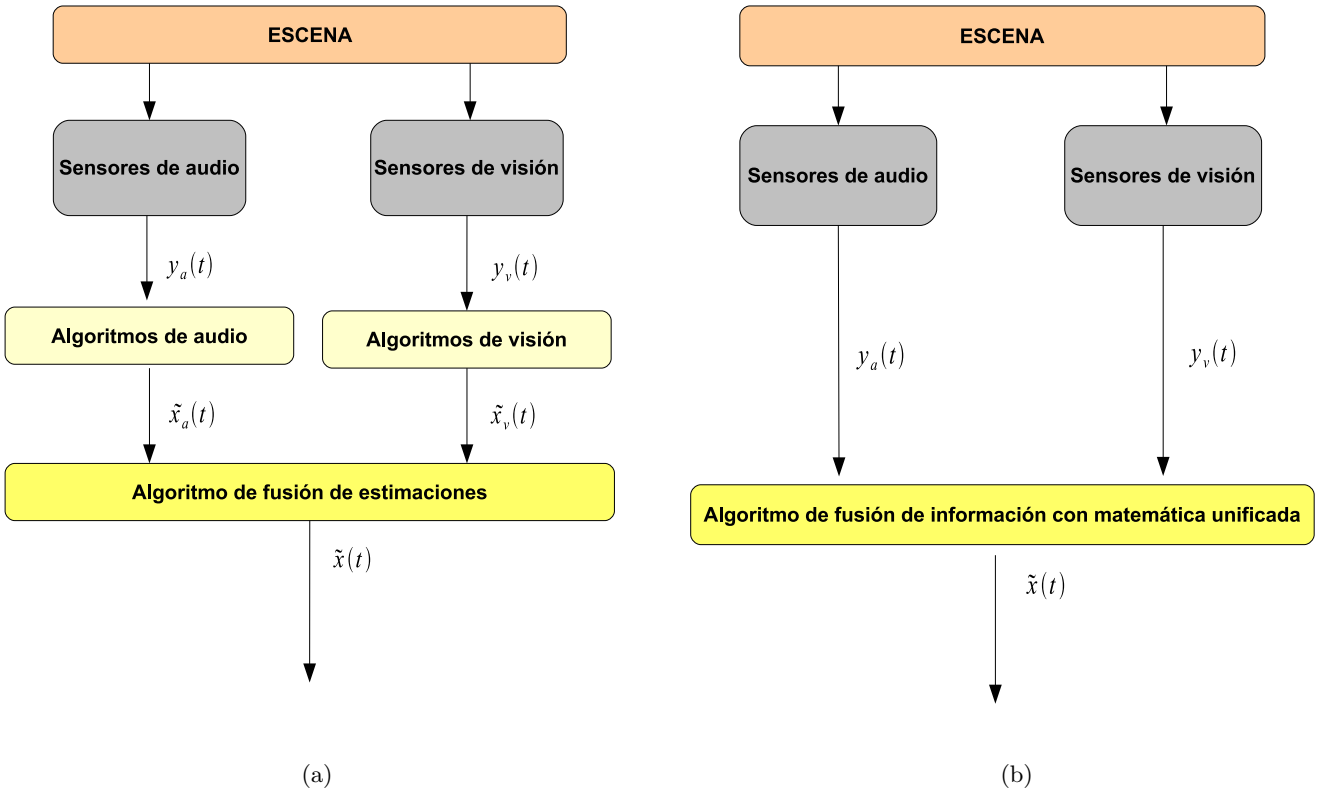


Figura 4.16: Clasificación de algoritmos de fusión audiovisual orientados a: (a) sistema, (b) modelo.

Figura 4.17: Modelo de observación para visión en [Vermaak, 2001]

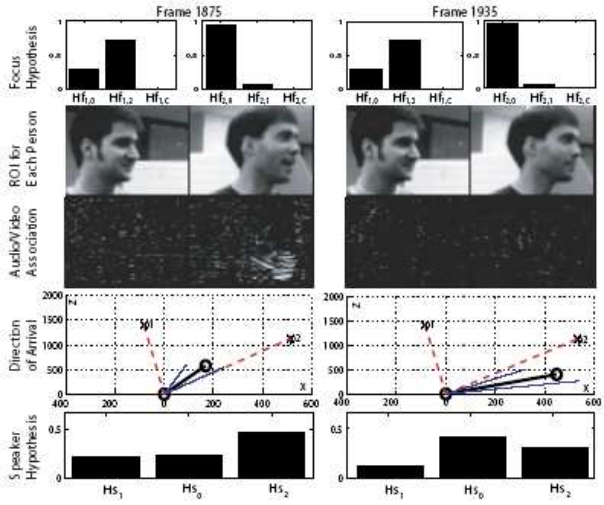


Figura 4.18: Resultados extraídos de los experimentos de [Siracusa, 2003]

Figura 4.19: Resultados de los experimentos de [Gatica, 2007]

la configuración de los sensores es bastante simple, con una única cámara y dos micrófonos, como el ejemplo mostrado en [67].

Más recientemente los trabajos se orientan ya a situaciones multi-persona con múltiples locutores activos y con configuraciones sensoriales más complejas [70] [71] (Figura 4.18) [72], [73] y [16] (Figura 4.19).

4.5. Métricas de evaluación

La evaluación CLEAR establece una serie de métricas para poder llevar a cabo una correcta evaluación de los resultados obtenidos al realizar los experimentos de localización de personas mediante información acústica. A continuación se muestran aquellas que han sido empleadas para evaluar los experimentos realizados en este trabajo.

- $Pcor = \frac{\text{nº error fine}}{\text{nº frames con estimación}}$ Teniendo en cuenta que los *error fine* son aquellos en los que el error en la estimación (distancia entre la localización verdadera y la estimación proporcionada) es menor que 500 mm.
- *Bias fine*: Distancia media en mm entre los *ground truth* (posiciones reales) y la estimación para los casos en que se genera un *error fine*. Muestra la habilidad del sistema de seguimiento para encontrar posiciones correctas.
- *Bias fine+gross*: Distancia media en mm entre los ground truth y la estimación tanto para los casos *fine error* como para los *gross error* (error entre la posición real y la estimación mayor de 500 mm).
- *AEE fine = MOTP (Multiple Object Tracking Precision)*: Parámetro que indica la precisión del sistema como la distancia euclídea en mm entre el *ground truth* y las estimaciones con *fine error* normalizada al número total de *fine errors*.

4.6. Conclusiones

Se han expuesto una serie de conceptos teóricos sobre la localización acústica, localización basada en información procedente de visión y localización basada en fusión audiovisual de manera detallada, siendo estos necesarios para la correcta comprensión de los desarrollos que se verán a continuación. Se está ahora en condiciones de pasar a explicar los algoritmos realizados, teniendo presente su base teórica.

Capítulo 5

Desarrollo algorítmico y herramientas

5.1. Introducción

En el algoritmo SRP de estimación de la posición de hablantes se han incluido ciertas modificaciones que se detallan en este capítulo, con el fin de mejorar los resultados obtenidos. Por otra parte, se exponen los aspectos que han permitido mejorar el sistema de experimentación basado en localización acústica con el objetivo de obtener un sistema más versátil.

Además, al sistema de experimentación basado en vídeo y desarrollado en [54] se le han añadido nuevas funcionalidades y mejoras que permiten realizar experimentos de una manera más sencilla. Por último, dicho sistema se ha convertido en una aplicación capaz de procesar información audiovisual y realizar estimaciones de la localización de los hablantes en función de información procedente de audio y visión, para lo cual han sido necesarios diferentes desarrollos que se detallan en este capítulo.

5.2. Diseño e implementación de mejoras en el sistema de localización acústica

La versión desarrollada en [8] previa a este trabajo contaba con diversas limitaciones y problemas, siendo dos de ellos los siguientes:

- Los ficheros de audio empleados como entrada del sistema sólo podían ser de tipo monocanal.
- La potencia de audio se calculaba empleando la aportación de cada par de micrófonos posibles entre los proporcionados al sistema.

Por ello se han desarrollado las siguientes dos mejoras en el algoritmo SRP con el objeto de:

- Permitir que el sistema funcione sobre los ficheros de audio de entrada tanto si son monocanal como multicanal.
- Dotar a SRP de la capacidad de agrupar el conjunto total de micrófonos en diferentes subarrays o agrupaciones, modificándose el cálculo de la potencia de audio de cada localización como se verá más adelante. De esta manera, en aquellas situaciones en las que existan micrófonos alejados que puedan introducir aliasing y errores en la localización, será posible agruparlos de tal forma que únicamente se tengan en cuenta las aportaciones de aquellos micrófonos más próximos entre sí.

5.2.1. Funcionamiento multicanal de SRP

Como se explica en el capítulo del Manual de Usuario de la página 139, se posibilita la utilización de ficheros de audio multicanal en el algoritmo SRP. En el proceso de experimentación existe un archivo denominado `audiosource` en el que se proporcionan diferentes variables, siendo destacables para la explicación de esta funcionalidad las siguientes:

- `NUMFILESPERUTT`: En cada seminario o locución grabada de cada base de datos son proporcionados una serie de ficheros de audio (de un único canal y varios). El número de ellos que hay que tener en cuenta para evaluar cada secuencia en concreto se proporciona en la variable `NUMFILESPERUTT`.

- Además, aparece un listado con los paths completos de todos los ficheros de audio a considerar mencionados anteriormente.

Por lo tanto, se emplean `NUMFILESPERUTT` ficheros de audio consecutivos del listado proporcionado en `audiosource` con el fin de llevar a cabo un experimento para un seminario en concreto. Dichos ficheros pueden ser de uno o varios canales, procesándose correctamente indicando, en caso de que existan archivos multicanal, los canales que se quieren leer de ellos. Estos canales se especifican en la variable `CHANNELS` proporcionada en los ficheros `go-GENDBLIS.cfg` y `go-SRP.cfg`, ver Manual de Usuario en la página 139.

A continuación se muestra el contenido de un archivo `audiosource`, donde se listan ficheros de audio monocanal y multicanal para dos seminarios distintos (LFD-P5-O1-U15004 y LFD-P5-O1-U01171):

```
   3 15
/usr/share/geintra/.../LFD-P5-O1-U15004-ch1.wav
/usr/share/geintra/.../LFD-P5-O1-U15004-ch2.wav
/usr/share/geintra/.../LFD-P5-O1-U15004.wav
/usr/share/geintra/.../LFD-P5-O1-U01171-ch1.wav
/usr/share/geintra/.../LFD-P5-O1-U01171-ch2.wav
/usr/share/geintra/.../LFD-P5-O1-U01171.wav
...
```

En este ejemplo la variable `NUMFILESPERUTT` adquiere el valor 3, por lo que para cada locución se utilizan 3 ficheros consecutivos del listado contenido en `audiosource`. Por ejemplo, para el seminario LFD-P5-O1-U15004 se leen dos archivos monocanal (canal 1 y 2) y uno multicanal del que se extraerán las señales del resto de canales a emplear en el experimento.

Para realizar este cambio se han modificado las funciones encargadas de leer los ficheros de audio, teniendo en cuenta su estructura interna, que se observa en la Figura 5.1. Como se puede ver se tiene una muestra para cada instante de tiempo; en el fichero monocanal se encuentran una tras otra por ejemplo para el canal 2, por lo que la lectura se realiza de la misma manera. Sin embargo en el fichero multicanal se encuentran las muestras de los diferentes canales para un mismo instante de tiempo de forma consecutiva, por lo que habrá que realizar saltos para acceder a aquellas muestras del canal de interés, en este ejemplo el canal 2.

5.2.2. Utilización de agrupaciones de micrófonos en el cálculo de la potencia en SRP

Como se ha comentado en el capítulo de Estudio Teórico en la página 33, para la estimación de la posición basándose en información de audio se emplean generalmente los arrays de micrófonos, debido principalmente ala mejora en la calidad de la señal recibida y a permitir una localización de objetos de forma efectiva.

Además, un problema también comentado es el *aliasing espacial*, fenómeno por el cual aparecen lóbulos no deseados en el patrón de directividad de los micrófonos, llevando esto a posibles errores en la localización. Cuanto menor sea la distancia entre los pares de micrófonos empleados en el experimento, menor probabilidad de que se produzca este fenómeno, y consecuentemente mejores resultados el proceso de estimación de la posición.

Por ejemplo, teniendo en cuenta que la voz humana se encuentra generalmente en el rango de 20 Hz - 20 KHz, la distancia máxima que podría existir entre los micrófonos estaría en el rango de 8.5 milímetros y 8.5 metros.

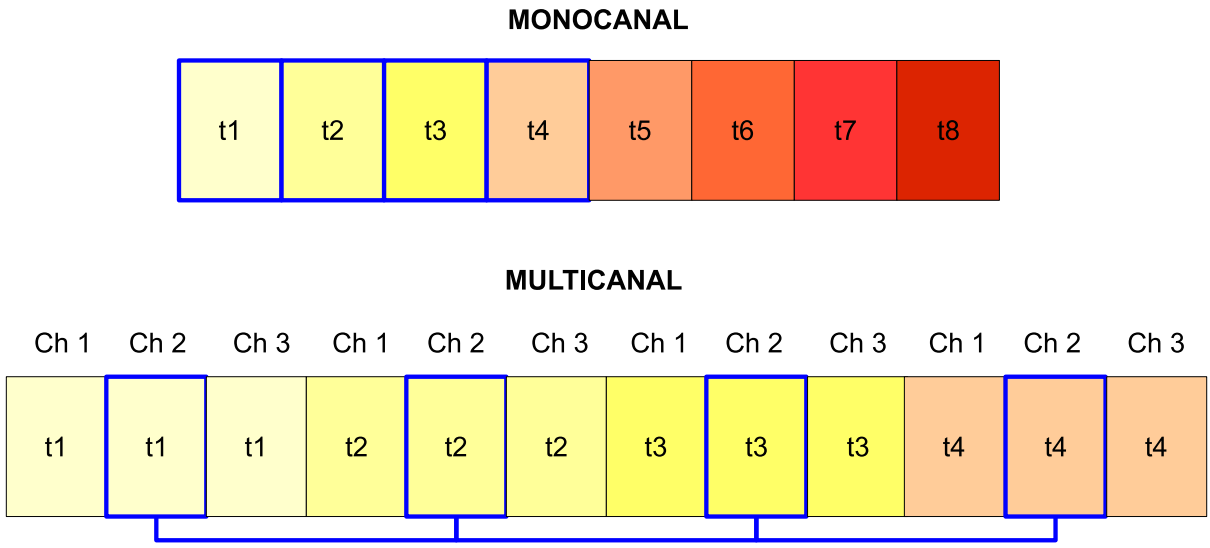


Figura 5.1: Estructura interna de un fichero de audio monocanal y multicanal con 3 canales

Para el cálculo de la potencia de audio de cada localización del espacio de búsqueda, se emplea la correlación de las señales de cada par de micrófonos existentes en la sala. La aportación que ofrecen dos micrófonos demasiado alejados no hará más que influir a la hora de generar *aliasing espacial*, empeorando los resultados.

Por tanto, se plantea la posibilidad de agrupar los micrófonos cercanos en subarrays, eliminando así la aportación de los pares de micrófonos más alejados en el espacio a la potencia total de audio de una localización determinada.

La potencia de cada punto se calculaba inicialmente como la suma de las aportaciones de potencia de cada par de micrófonos existentes en la sala, emparejando cada micrófono de cada array con el resto de micrófonos. Con esta modificación de la agrupación en subarrays, la potencia de cada localización se calcula como la suma de la aportación únicamente de cada subarray, siendo ésta a su vez la suma de la aportación de los pares de micrófonos sólo de dicho subarray:

$$P[q] = \sum_{i=0}^{i=n} P[i][q] \text{ donde } q \text{ es una localización espacial dada y } n \text{ es el número de subarrays} \tag{5.1}$$

$$P[i][q] = \sum_{j=0}^{j=m} P[j][q] \text{ donde } m \text{ es el número de pares de micrófonos del subarray} \tag{5.2}$$

Para mostrar el efecto de esta modificación se agrupan en subarrays los micrófonos de un experimento de test de la base de datos AIT de CLEAR 2007, cuya sala se muestra en la Figura

5.2. En la Figura 5.3 se observa la tipología de los arrays de micrófonos, existiendo 4 en la sala: A, C, D y MK3 en la Figura 5.2.

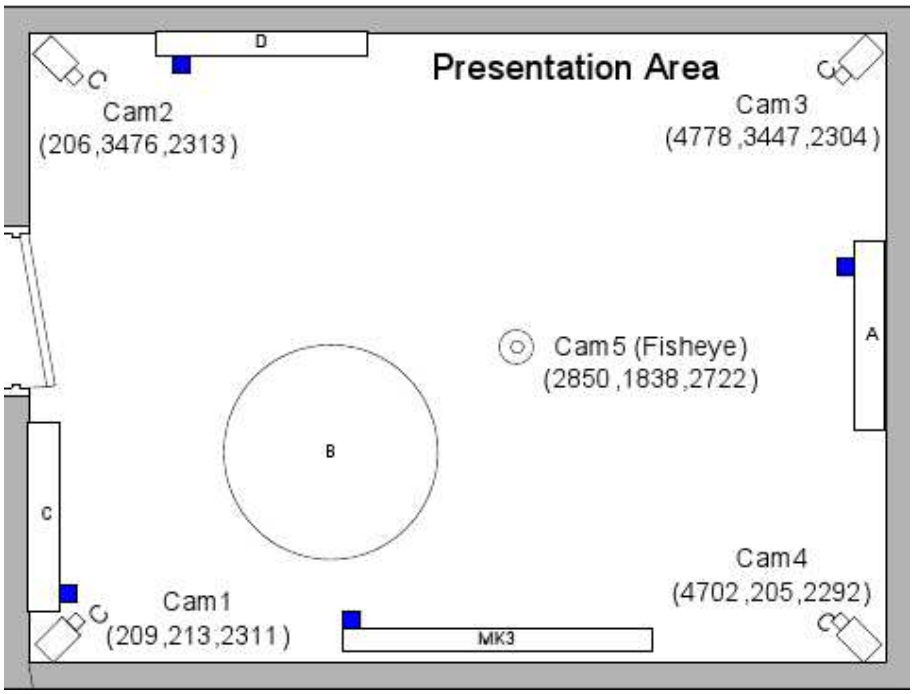


Figura 5.2: Sala AIT

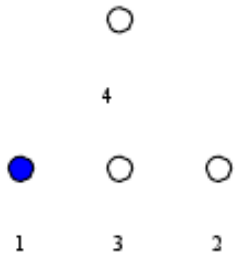


Figura 5.3: Sala AIT

Se tienen en cuenta para estos experimentos los micrófonos pertenecientes a los arrays de 4 micrófonos: A, C y D en la Figura 5.2.

En un primer experimento no se ha realizado la agrupación de micrófonos en subarrays, teniéndose en cuenta entonces todos los pares posibles de ellos para el cálculo de la potencia de cada localización. En la primera columna de la Tabla 5.1 se muestran los resultados obtenidos. En un segundo experimento se establecen 3 subarrays, cada uno de ellos correspondiendo a cada array de micrrófonos; por lo que la aportación de los pares de micrófonos formados por ejemplo por un micrófono del array A y otro del C no se tendrían en cuenta. Los resultados se observan en la segunda columna de la Tabla 5.1. Por último, en la tercera columan de la Tabla 5.1 se repite el experimento con 6 subarrays, pero en esta ocasión en cada subarray se mezclan dos micrófonos de arrays distintos, por lo que es de esperar que los resultados sean desfavorables.

Como se ha visto en la sección Métricas de evaluación de la página 65, un *fine error* es una buena estimación ya que el error entre la posición estimada y la real es menor que 500 mm, por

Tabla 5.1: Resultados obtenidos con diferentes configuraciones de micrófonos en AIT

	Sin subarrays	Con 3 subarrays	Con 6 subarrays
Pcor	$30{,}0 \pm 7{,}7\,\%$	$59{,}0 \pm 8{,}3\,\%$	$9{,}0 \pm 4{,}8\,\%$
Rel. error reduction		$96{,}7\,\%$	$-70{,}0\,\%$
Bias fine (x:y:z) [mm]	$-73 : 92 : -91$	$-13 : -27 : -102$	$17 : -95 : -187$
Bias fine+gross (x,y,z) [mm]	$-268 : -78 : -184$	$-47 : -142 : -146$	$-138 : -589 : -256$
AEE fine [mm] = MOTP	293	233	396
Rel. AEE reduction		$20{,}5\,\%$	$-35{,}2\,\%$
Fine+gross [mm]	1017	585	1504
Rel. BIAS f+g reduction		$42{,}5\,\%$	$-47{,}9\,\%$
Loc. frames	135	135	135
Ref. duration (s)	309,0	309,0	309,0

lo tanto cuanto mayor sea el número de este tipo de errores, mejores resultados se obtienen. El parámetro Pcor es el más significativo ya que indica la cantidad de *fine errors* en el experimento. Se observa que sin agrupación de micrófonos Pcor tiene un valor del 30 %, mientras que con 3 subarrays se pasa al 59 %, suponiendo esto una mejora de casi el 97 %. En la última columna, donde se han agrupado micrófonos de distintos arrays en 6 subarrays se obtiene una tasa muy baja, disminuyendo el valor de Pcor en un 70 % respecto al primer caso.

Agrupando los micrófonos en este experimento se produce una gran mejoría en los resultados, debido principalmente a que los arrays de la sala se encuentran alejados, como se puede ver en la Figura 5.2.

5.3. Diseño e implementación de un nuevo sistema de experimentación para el sistema de localización basado en audio

El sistema de experimentación desarrollado en [8] era un sistema mediante el cual lanzar un nuevo experimento se convertía en una tarea tediosa y básicamente manual, sobre todo en el momento de ejecutar un experimento de una nueva base de datos. Ya que se desea llevar a cabo multitud de experimentos surge la necesidad de mejorar este sistema, automatizando en todo lo posible las tareas de cada fase.

Para ello se ha desarrollado un conjunto de scripts que permiten la generación de experimentos de manera automática, modificando únicamente parámetros de configuración necesarios, ver la sección Manual de Usuario de la página 139.

Los objetivos que han sido abordados en la realización del nuevo sistema de experimentación son:

- Implementar un sistema de experimentación versátil que cubra todas las etapas necesarias en la realización del experimento.

- Generar de manera sencilla un nuevo experimento.
- Modificar únicamente los parámetros de configuración para generar un nuevo experimento.
- Facilitar al usuario el seguimiento al recibir información de los pasos que se están realizando.
- Convertir en una tarea sencilla la modificación del código ya existente para añadir nueva información.
- Dotar de transparencia a la mayoría de procesos internos de cara al usuario, no siendo necesaria su total comprensión para ser capaz de realizar un nuevo experimento.

Mediante la implementación del nuevos sistema de experimentación se ha conseguido:

- Generar un sistema de experimentación automático y sencillo que cubre las etapas de configuración, ejecución y evaluación.
- Facilitar la generación de un nuevo experimento mediante un sistema de automático, donde únicamente hay que proporcionar el nombre del experimento.
- Restringir los parámetros de configuración de cada etapa a su respectivo script de configuración.
- Generar de forma automática de un archivo con el listado de los ficheros de audio involucrados a partir del respectivo fichero de configuración.
- Ejecutar de forma sencilla el algoritmo SRP, siendo únicamente necesario revisar unos parámetros en el script de configuración correspondiente.
- Generar de manera automática el paso de evaluación, que tiene asociado un script de ejecución y otro de configuración, al igual que el resto de etapas.
- Generar una serie de logs con los pasos llevados a cabo en las etapas de ejecución y evaluación, inlcuyendo del mismo modo información sobre errores producidos, y siendo especialmente útil a la hora de depurar nuevas utilidades añadidas.
- Ofrecer comodidad añadida al ser enviado un e-mail al usuario que ha lanzado el experimento cuando la etapa de ejecución del SRP (generalmente larga) ha terminado, informando si ha concluido con éxito o con algún error.
- Facilitar el proceso de adición de una nueva base de datos.
- Conservar los resultados de evaluación obtenidos y los logs procedentes de múltiples ejecuciones de un mismo experimento y localizar de manera sencilla los mismos ya que se antepone fecha y hora a todos ellos. Esto es especialmente útil cuando se está buscando la configuración de parámetros que arroje los mejores resultados de evaluación.

5.4. Diseño e implementación de mejoras en el sistema de localización basado en visión

El sistema de experimentación basado en visión llevado a cabo en [54] contaba con multitud de limitaciones a la hora de cambiar las características del tipo de experimento a realizar:

- Matriz de homografía no calculada por el propio software, sino introducida manualmente.
- Imposibilidad de lanzar más de un servidor desde un mismo host.
- Inexistencia de parámetros de entrada, estando todos ellos definidos en los propios ficheros.
- Imposibilidad de tratar imágenes de tamaño diferente a 640x480 píxeles.
- ...

Por todo ello, al sistema de localización desarrollado en [54] se le han añadido una serie de prestaciones que mejoran el sistema, añadiendo nuevas funcionalidades y mejorando las ya existentes, llegando a mejorar su versatilidad, donde el usuario de manera sencilla lanza nuevos experimentos.

A continuación se explican, una a una cada prestación o mejora implementada:

5.4.1. Mejoras implementadas en la aplicación servidor

- Para la ejecución de un servidor es necesario proporcionar una serie de parámetros de entrada de tal forma que el usuario no deba modificar nada en el código a no ser que vaya a realizar un experimento referente a una nueva base de datos, o cambie algún parámetro de los no especificados por línea de comandos, no siendo muy usual.
  En los parámetros de entrada (explicado con detalle en el Manual de usuario de la página 139) se aporta información sobre:
  - Fichero `.ini` de calibración.
  - Si el `.ini` contiene una matriz de homografía o parámetros de las cámaras.
  - Si el experimento es o no de audio.
  - El fichero de potencias en caso de ser un experimento basado en información acústica.
  - Tamaño del píxel en milímetros.
  - Offset para calcular la homografía.
  - Offset del servidor.
- Como se ha comentado anteriormente, es necesario el cálculo de una matriz de homografía para poder proyectar una imagen capturada por una cámara sobre un plano dado. Para ello el sistema detecta automáticamente si la información del archivo `.ini` corresponde a los parámetros de calibración de las cámaras o directamente a la matriz de homografía. En caso de encontrarse parámetros de calibración se calcula de manera automática la matriz de homografía, para lo cual son necesarios ciertos parámetros como el tamaño del píxel o el offset respecto a la posicicón (0,0,0) al que se quiere calcular dicha matriz, todos ellos proporcionados por línea de comandos mediante la llamada a la aplicación servidor.
  Además, los parámetros de calibración suministrados corresponden a un tamaño de imagen determinado, generalmente 640 de ancho y 480 de alto (píxeles). En cambio, el tamaño de las imágenes manipuladas tanto en cliente como servidor es de 320 por 240, por lo que se hace necesario un redimensionamiento de la matriz de homografía que se lleva a cabo de forma atutomática.
  En los experimentos CHIL siempre se proporcionan los parámetros extrínsecos e intrínsecos de las cámaras.
  En la Figura 5.4 se muestra los parámetros de calibración de las cámaras, la matriz de homografía calculada y su posterior redimensionamiento.

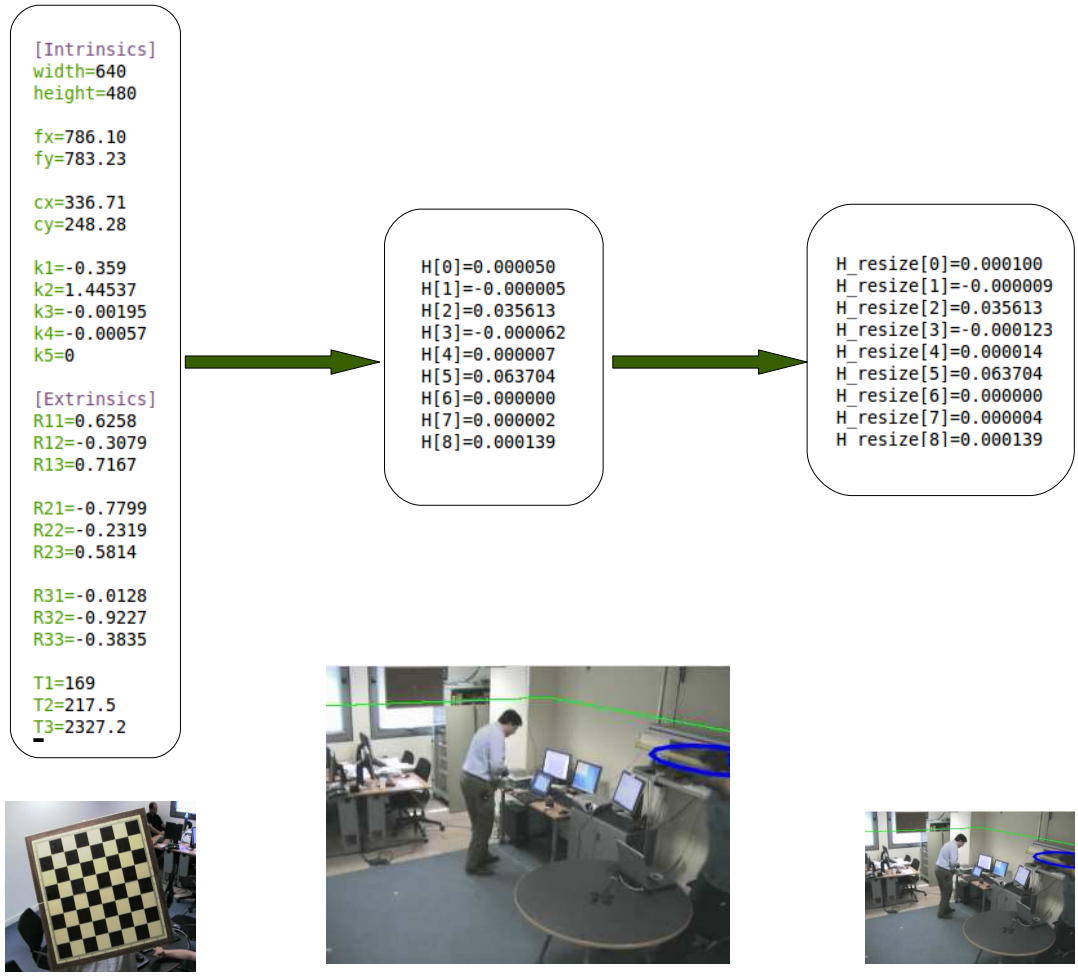


Figura 5.4: Cálculo de la matriz de homografía y redimensionamiento

- La aplicación desarrollada es capaz de funcionar tanto en tiempo real como en modo simulación, es decir, con datos procedentes de las bases de datos disponibles, e indicado a través de una constante denominada `SIMULACION`. Cuando se trabaja en tiempo real intervienen las cámaras del Espacio Inteligente, y por tanto es necesaria una correcta configuración de las mismas así como un envío de imágenes capturadas de la cámara al cliente, para lo que se emplean las funciones del módulo `raw1394`. En cambio, al trabajar en modo simulación todas estas funciones son ignoradas ya que las imágenes se cargan directamente de los repositorios de las bases de datos. El algoritmo está preparado para adaptar el funcionamiento en modo simulación a cada experiento y cada base de datos empleada, debiendo ser modificadas variables referentes a nomenclatura de las imágenes, del background, etc.

- Se ha conseguido que el código fuente sea más versátil y escalable mediante la declaración de varias constantes influyentes en el comportamiento del sistema.

- Cuando se trabaja en modo simulación se emplea información de diferentes bases de datos CLEAR, la cual por comodidad se almacenan en un mismo pc. De esta manera, todos los servidores deben lanzarse desde el mismo host para que tengan acceso a la información de la base de datos: imágenes, posiciones etiquetadas, background, etc. Esto es posible gracias al parámetro offset del servidor, que caracteriza de forma única a los servidores, se hayan lanzado desde un mismo o varios hosts.

- Cuando se ejecuta un experimento es deseable conocer en tiempo real la calidad de los resultados obtenidos. Para ello se representan gráficamente en las imágenes de los servidores las posiciones etiquetadas por CLEAR (coordenadas x, y, z de la posición real de las personas) , comprobando así si la estimación realizada por nuestro sistema es correcta o no. CLEAR proporciona etiquetado de posiciones reales de las personas que se encuentren hablando en la escena, de aquellas que simplemente aparecen en las imágenes o de ambas a la vez. Por lo tanto es posible representar dichas posiciones tanto para las imágenes generadas a partir de información acústica como aquellas procedentes de visión.

- El sistema de visión, funcionando en tiempo real, necesita una cantidad de frames iniciales para calcular el fondo de la imagen y así poder realizar la resta de dicho fondo a todas las imágenes capturadas a partir de ese instante, método empleado para localizar los objetos. De la misma forma, y para el correcto funcionamiento de la aplicación, cuando se trabaja en modo simulación es igualmente necesario disponer de un background con imágenes del fondo de la sala. Además, CLEAR proporciona imágenes de background para todas ellas. Se ha añadido dicha funcionalidad que permite establecer unos frames para el cálculo del fondo (denominados `BACKGROUND`) trabajando con las bases de datos CLEAR. Durante '`BACKGROUND`' frames se cargan imágenes del fondo de la sala, permitiendo al sistema calcular el fondo de la misma. A partir de ese instante se lleva a cabo la resta del fondo calculado a las imágenes cargadas, de tal manera que se procede a la localización de objetos en escena.

- Los servidores y el cliente intercambian continuamente información. Mediante una serie de comandos recibidos, se especifica a cada uno de ellos las acciones a realizar. Se ha añadido un nuevo comando mediante el cual cada uno de los servidores envía información de interés al cliente:

  - Nombre del fichero `.ini` para el cálculo de la homografía.
  - IP del servidor, incluyendo el offset del mismo.
  - Ancho y alto de las imágenes del servidor en píxeles.

- Matriz de homografía.
- Matriz de homografía inversa.
- Variable que especifica si el servidor es de audio o vídeo.

5.4.2. Mejoras implementadas en la aplicación cliente

- El sistema funciona ahora correctamente independientemente del tamaño del píxel empleado, tanto para la coordenada x como y. El valor del mismo se establece mediante una constante definida en un fichero de configuración.
- Es posible almacenar la información devuelta por el Filtro de Partículas en un fichero de salida, guardando la posición x, y, z en milímetros de las clases validadas así como su identificador. Esta información es almacenada con el formato CLEAR, permitiendo así utilizar este fichero para realizar una posterior evaluación de los resultados obtenidos.
- En la aplicación visual del cliente, los servidores tienen ahora el formato IP:offset permitiendo así la conexión con varios servidores lanzados desde un mismo host. Si no se especifica ningún offset se asume como 0.

5.5. Diseño e implementación de un sistema de localización basado en fusión audiovisual

5.5.1. Aplicación cliente-servidor

La metodología seguida en el trabajo desarrollado en [54] y empleado como punto de partida para realizar el sistema de experimentación basado en fusión audiovisual pasa por utilizar una arquitectura cliente-servidor, donde cada cámara es tratada como un servidor que sirve imágenes al cliente, siendo este último el encargado de tratar y mostrar la información. Los obstáculos son presentados en un grid de ocupación de dimensiones y resolución prefijada.

Para el intercambio de datos entre los distintos sistemas se ha optado por la arquitectura cliente-servidor, donde es el cliente el que se conecta a los servidores y hace peticiones a éstos, ver Figura 5.5 [54].

5.5.2. Formación de la imagen de audio

Con el fin de integrar el sistema de experimentación basado en audio en aquél desarrollado en [54] se desea construir una imagen partiendo de información acústica, para tratarla así como si procediera de una cámara más y realizar el seguimiento basándose no sólo en la información procedente de visión.

Como se puede observar en el Manual de Usuario de la página 139, uno de los ficheros de salida que genera la aplicación SRP es el `baseName.pow`, en el que se proporciona para cada instante de tiempo y cada punto de búsqueda del espacio, la potencia de audio correspondiente.

Si en un instante determinado se convierte el valor de potencia para cada uno de los puntos del espacio a un valor equivalente en escala de grises, es posible generar una imagen o mapa de potencias. De esta forma, los puntos con potencia 0 se representan de color negro, y los puntos que posean la potencia máxima del frame representan los puntos en color blanco. Con

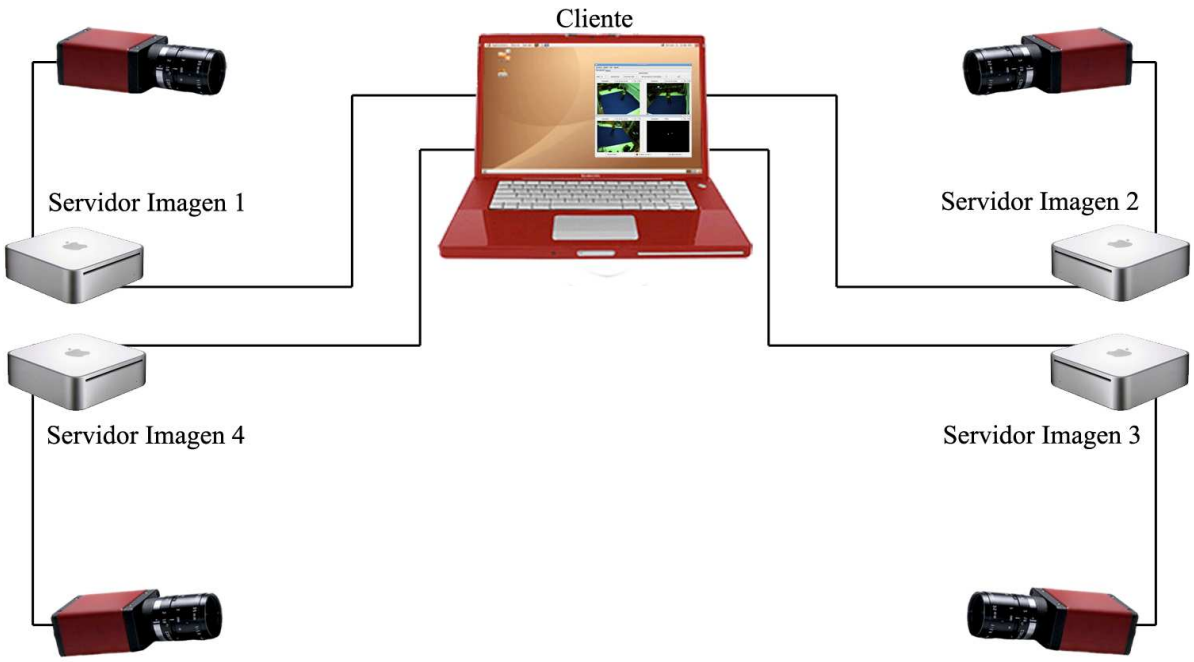
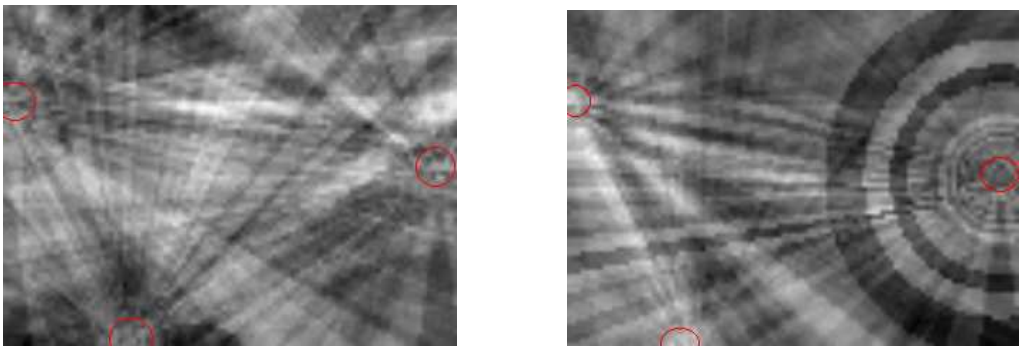


Figura 5.5: Arquitectura cliente-servidor

esta asignación de color del píxel se obtienen imágenes como las mostradas en la Figura 5.6, donde se encuentra marcados con círculos rojos la posición de los arrays de micrófonos.

El sistema de experimentación basado en audio tiene un funcionamiendo 2D, por lo que la información procedente de audio debe ser proporcionada en dos dimensiones de la misma manera. Así, se ha generado un fichero de potencias estableciendo la coordenada z correspondiente a la altura a un valor de 1700 mm, ya que se considera la altura media de una persona que se encuentre de pie. Por lo tanto, se tiene la potencia para todo x y todo y de dicha altura.

En la Figura 5.2 se muestra la sala correspondiente al experimento del cual se han obtenido las imágenes mostradas y en la Figura 5.7 se puede ver de forma superpuesta la imagen de potencias generada sobre la sala en la que se ha realizado la grabación. Hay que aclarar que el eje de coordenadas y de la imagen se encuentra invertido con respecto al mismo en el sistema de experimentación; y que se representan en color azul todos los micrófonos, habiéndose usado únicamente 4 arrays micrófonos para el experimento mostrado (A, C y D en la figura).



(a) (b)

Figura 5.6: Formación de una imagen partiendo de un fichero de potencias en dos situaciones: (a) sólo con ruido de fondo (b) ruido de fondo y un ruido localizado

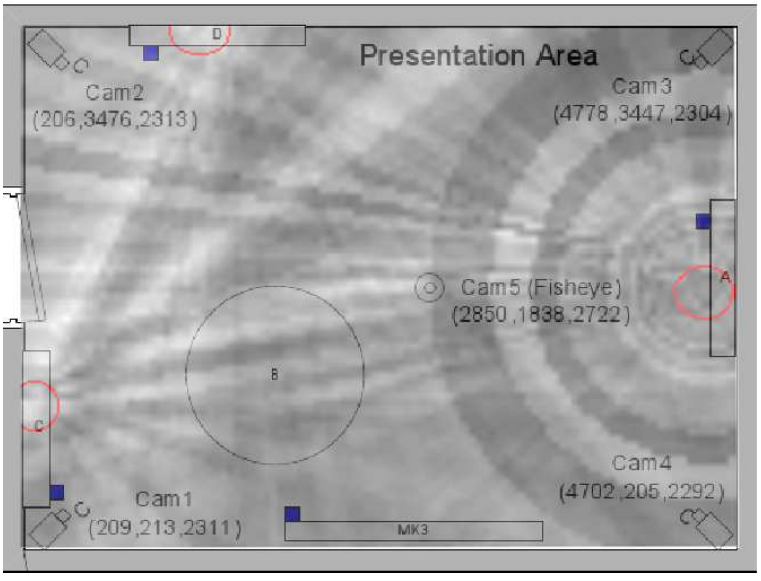
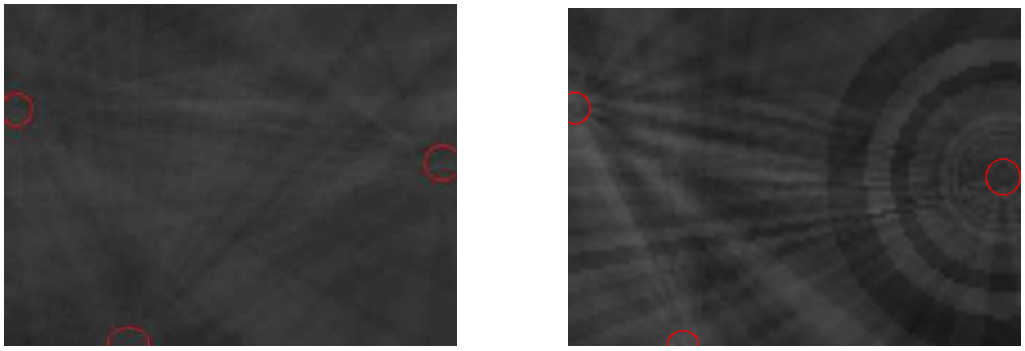


Figura 5.7: Mapa de potencias de un experimento concreto en la sala AIT

La imagen 5.6(a) corresponde a una situación donde únicamente existe ruido de fondo, mientras que en la 5.6(b) se localiza un ruido (toc toc de una puerta) en la parte izquierda superior

de la imagen. En ambas imágenes se he llevado a cabo un escalado por frame, es decir, los píxeles con potencia máxima del frame representan el valor 255 en escala de grises mientras que aquellos con potencia mínima el valor 0. Como se puede observar no ofrecen información demasiado clara, ya que en ambas imágenes aparecen zonas blancas (potencia alta) y más oscuras (potencia baja).

En los momentos en los que se presentan ruidos localizados o señal de habla la potencia es más alta; mientras que en los instantes donde sólo existe ruido de fondo el valor de dicha potencia será menor. Teniendo en cuenta que en la mayor parte del experimento existe algún tipo de señal de habla, resulta un dato de interés conocer el rango en el que se pueden agrupar los valores de la potencia de audio para el experimento completo. De este modo, se calcula la distribución de potencias, pudiendo así observar en qué franja se localizan los valores que aparecen más a lo largo del experimento. Se obtiene pues un rango de interés y se descarta la información procedente de aquellos puntos cuya potencia no supere el umbral mínimo, estableciendo el valor del píxel correspondiente a 0. Además, queda establecido un valor máximo, por lo que aquellos puntos que presenten potencia mayor que el máximo dispondrán de un valor en escala de grises de 255. Por ejemplo en un experimento se comprueba que los valores de potencia se agrupan entre 1.4 y 8.6 (escalados todos los valores de potencia entre 0 y 10), por lo que aquellos píxeles con potencia menor de 1.4 se representan en negro (0 en escala de grises), mientras que en los que su potencia es mayor que 8.6 (255 en escala de grises) aparecen en color blanco. Los píxeles cuya potencia se encuentre entre el máximo y el mínimo se representan con su nivel de gris correspondiente. Se consigue así que los frames en lo que no haya señal de habla o sonidos aparezcan de forma más oscura, encontrándose píxeles con valores más elevados en localizaciones con potencia de audio mayor. En la Figura 5.8 se muestran las mismas escenas que en la Figura 5.6 pero representedas con el nuevo rango de potencias válido.



(a) (b)

Figura 5.8: Misma escena con umbralización de la imagen basada en potencias: (a) sólo con ruido de fondo (b) ruido de fondo y un ruido localizado

Como se puede observar, supone un problema el hecho de formar un grid a partir de estas imágenes, por lo que se propone una solución alternativa para obtener información de interés, explicada a continuación:

- Se parte de las imágenes sin umbralizar escaladas entre el máximo y el mínimo del frame, como por ejemplo las mostradas en 5.6.
- Se buscan los píxeles con valores máximos de potencia del frame en cuestión, aquellos cuyo nivel de gris supere el valor 250. Este valor se ha establecido experimentalmente de tal forma que se represente suficiente información sobre los puntos con níveles máximos de potencia.

- Se dilatan los puntos máximos obteniendo una forma de mayor tamaño.
- Se genera una imagen con aquellos puntos que superen el valor de nivel de gris 200, valor escgido de forma experimental.
- Se realiza una AND de la imagen con los puntos dilatados con la imagen umbralizada a 200, de tal forma que los puntos adquieran su forma originaria.

Este proceso se muestra a continuación en imágenes:

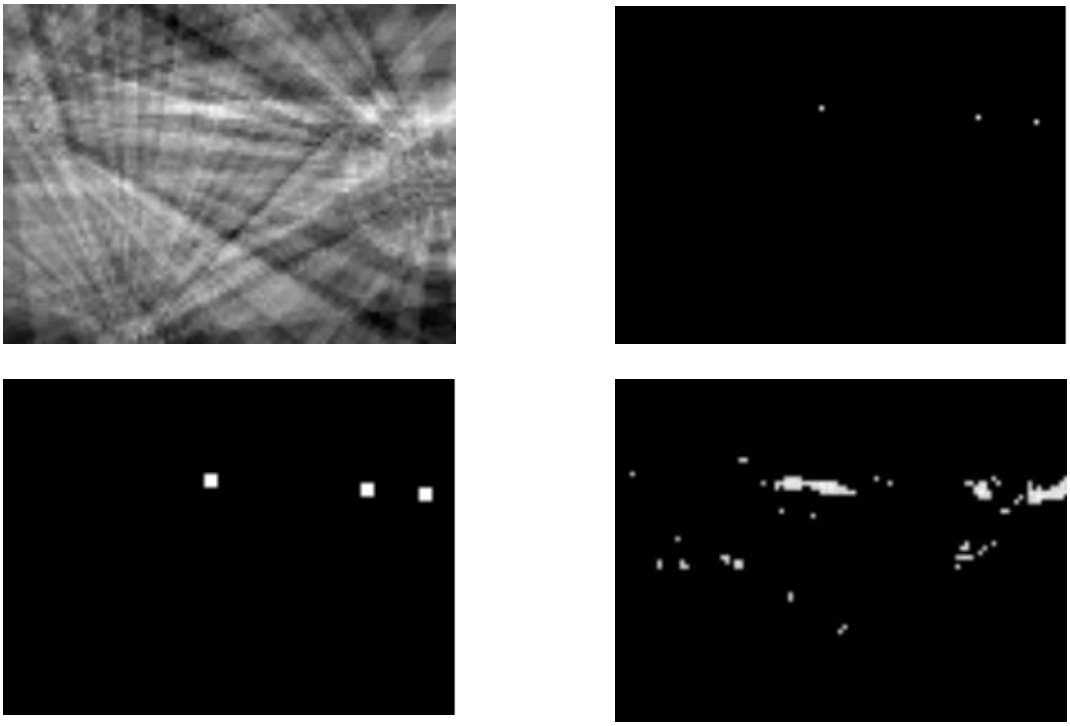


Figura 5.9: Arriba izquierda: imagen original. Arriba derecha: máximos de la imagen. Abajo izquierda: Puntos máximos dialatados. Abajo derecha: Imagen original filtrada a un nivel de gris de 200

Figura 5.10: (e) Resultado de realizar la AND entre la imagen con los puntos dilatados y la filtrada a 200

Se tiene por tanto una imagen con aquellas zonas donde la potencia es elevada representadas de color blanco, mientras que el resto de puntos se mantienen en negro.

Hay que destacar que esta estrategia está por validar, siendo una posible línea futura de trabajo a este proyecto.

5.5.3. Aplicación de la matriz de homografía

Una vez que se dispone de una imagen con información sobre los puntos donde existe señal de voz o ruidos localizados es necesario aplicarle la matriz de homografía inversa, que proyecta el plano imagen sobre el plano de proyección.

Como se comentó anteriormente, se ha generado una imagen con los valores de potencia para todos los puntos de la sala en cuestión a una altura prefijada y constante. Esta situación simula la captura de una cámara que se encontrara en el techo de la habitación, enfocando al suelo y centrada por completo en la sala.

Por lo tanto, la imagen calculada es directamente su proyección en el plano de representación (plano del suelo), ya que dicha proyección sobre *z=0* es la misma imagen obtenida. En la Figura 5.11 se observa esta transformación.

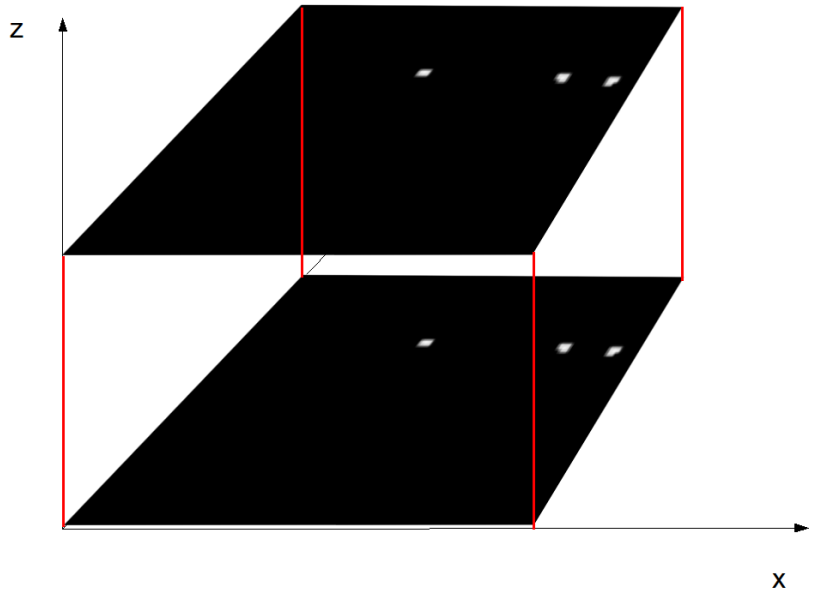


Figura 5.11: Imagen obtenida para altura z=1700 mm y su proyección sobre el plano del suelo

Viendo esta figura, cabe esperar que no es necesario aplicar ninguna tranformación, y por tanto tampoco emplear ninguna matriz de homografía. Sin embargo, es indispensable aportar dicha matriz ya que el sistema está preparado para ello y en caso de no proporcionarla se reporta un mensaje de error. La matriz de homografía inversa necesaria para proyectar directamente los puntos sobre el plano *z=0* es la siguiente, siendo la matriz identidad:

$$H^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3}$$

5.5.4. Obtención del grid de audio

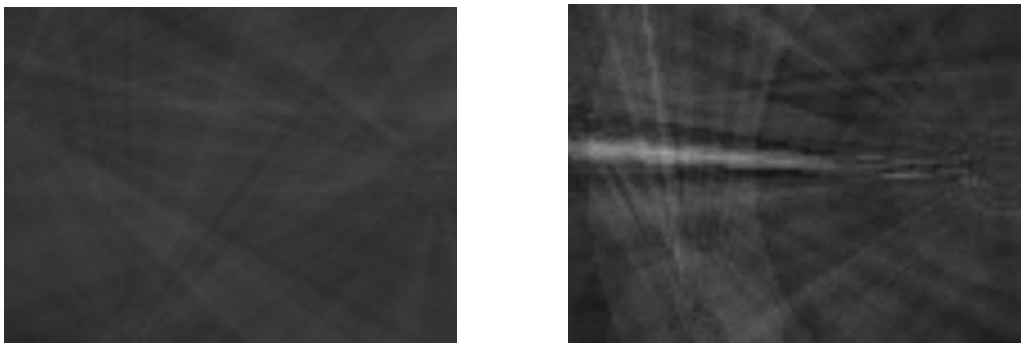
Hasta este momento se dispone de una imagen que muestra en color blanco aquellos puntos de mayor potencia para cada frame. Según esto, en cada frame existiría una fuente de voz o

algún tipo de ruido cuya potencia es suficientemente elevada. Pero esto realmente no es así, sino que en gran parte de los instantes únicamente aparece ruido de fondo, mientras que en algunos de ellos existe una señal de habla como tal.

Si se contara con un detector VAD (*Voice Activity Detector*) el problema estaría solucionado, pues mediante este sistema se conoce en cada instante si hay actividad de voz o no. Al no disponer de un VAD, y para solventar este problema se hace necesario por tanto un proceso que valide o no el grid obtenido, proporcionando un grid final donde sólo en aquellos casos en los que exista voz o ruidos se muestren puntos blancos en su determinada localización. Dicho proceso se pasa a explicar detalladamente a continuación.

En primer lugar se desea caracterizar de algún modo los frames de silencio, entendiéndose como tal aquellos en los que no existe otra perturbación salvo el ruido de fondo. Para ello se analiza la grabación disponible del experimento en cuestión en busca del fragmento donde sólo se encuentre silencio. Una vez hecho interesa analizar el fragmento con mucha precisión, por lo que se lanza un experimento SRP únicamente para el fragmento seleccionado, con un valor de `frame shift` pequeño, (el `frame shift` está involucrado en la velocidad de generación de muestras).

Previamente se ha observado que la variabilidad en el nivel de gris de los píxeles cuando hay 'silencio' es menor que en aquellos momentos en los que aparece alguna fuente de ruido, ver Figura 5.12 (imágenes con umbralización). Por lo tanto, si se calcula la desviación típica en el experimento en el que sólo existe silencio se podrá validar el grid final siempre que la desviación típica del frame sea mayor que la calculada en esos casos de silencio.



(a) (b)

Figura 5.12: Variabilidad del nivel de gris de los píxeles en dos situaciones: (a) silencio, (b) fuente de ruido.

En la Figura 5.13 se muestra la distribución de probabilidad de la desviación típica del nivel de gris de los píxeles calculada en el experimento donde no aparece ninguna fuente de ruido excepcional, únicamente ruido de fondo o silencio. Se observa que responde prácticamente a una distribución de probabilidad normal. Se puede asumir entonces que en el intervalo $[\mu-3\sigma, \mu+3\sigma]$ se encuentra comprendida, aproximadamente, el 99,74 % de la distribución. Por lo tanto, se validará el grid de aquellos frames cuya desviación típica de nivel de gris se encuentre fuera de este rango.

El procedimiento de este sistema de validación es el siguiente:

- Se analiza previamente el fichero de audio del experimento a realizar, buscando el fragmento inicial donde sólo aparece ruido de fondo.
- Se establece ese tiempo inicial como *background* de la siguiente manera: $frames_{background} =$

$fps * tiempo_{inicial}$ Donde fps son los frames por segundo a los que se realiza el experimento.

- Cuando comienza el experimento, durante los frames pertenecientes al *background* se calcula la distribución de probabilidad de la desviación típica del nivel de gris de los píxeles (media y desviación típica de la desviación típica del nivel de gris).
- Durante este tiempo, el grid enviado al cliente es una imagen completamente negra.
- Una vez que se supera el tiempo de *background*, el grid calculado se valida si la desviación típica del nivel de gris se encuentra fuera del rango de $[\mu - 3\sigma, \mu + 3\sigma]$.
- El grid validado se envía al cliente, por el contrario, si el grid no se valida se envía al cliente una imagen con todos los píxeles en color negro.

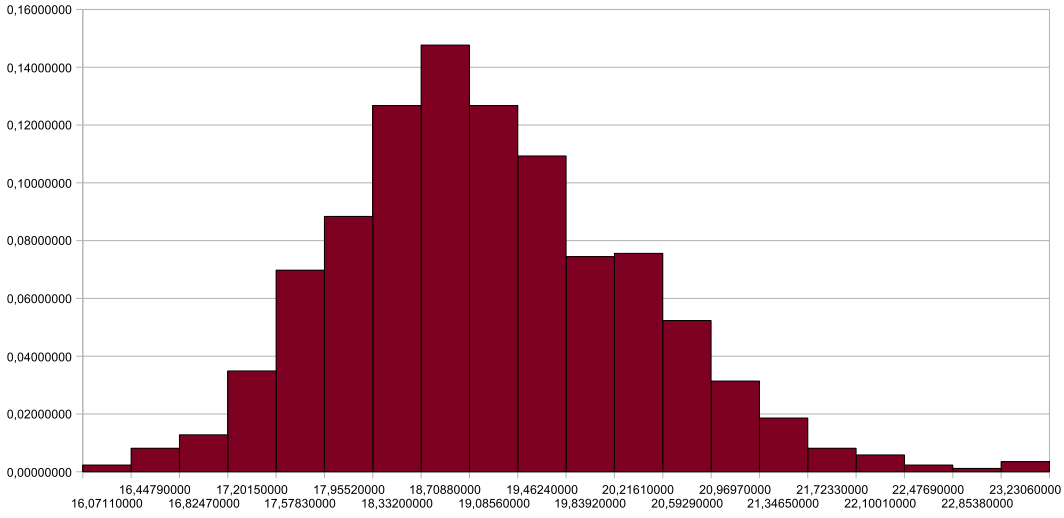


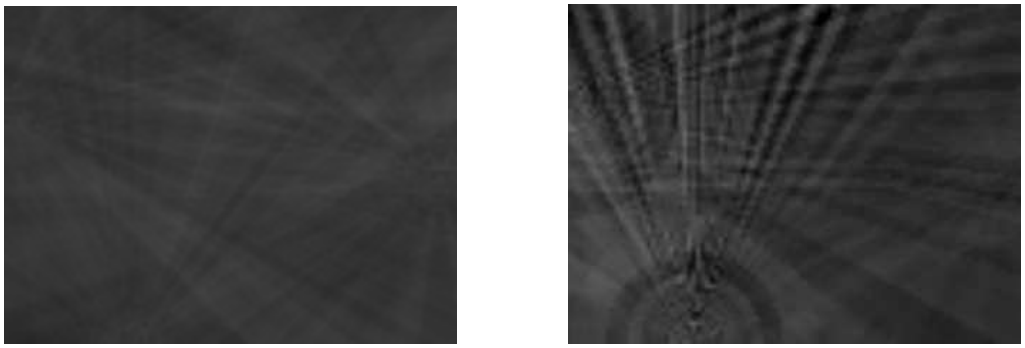
Figura 5.13: Distribución de probabilidad de la desviación típica de los píxeles en el experimento de silencio

A continuación se muestra un ejemplo concreto para el seminario AIT de CLEAR 2007. La desviación típica de los frames iniciales con silencio tiene una media $\mu = 5{,}591$ y una desviación típica $\sigma = 1{,}803$, por lo que se valida el grid de aquellos frames cuya desviación típica sea mayor que $\mu + 3\sigma = 11$. La imagen 5.14(a) tiene una desviación típica de 4.903 por lo que su grid no se envía al cliente; sin embargo, la desviación de la imagen de la Figura 5.14(b) tiene un valor de 14.319, el grid generado es validado y enviado al cliente.

La estrategia seguida como método de validación del grid de audio está pendiente de validar, por lo tanto se propone como posible línea futura de trabajo a seguir a partir de este proyecto.

5.5.5. Obtención del grid procedente de la fusión audiovisual

En este punto se tiene un grid procedente del sistema de audio y un grid procedente del sistema de visión.



(a) (b)

Figura 5.14: El grid generado a partir de la imagen (a) no se valida, el correspondiente a la imagen (b) sí

Para obtener un grid final constituido por la fusión audiovisual se ha "sumado" dicha información, es decir, se ha realizado una OR lógica con las imágenes del grid de visión y audio. El grid de audio ofrece información adicional, conociendo así la posición estimada de las fuentes de audio que aparecen en el experimento.

Prácticamente en todo momento existen personas en la escena de análisis, por lo que se dispone de información procedente de visión en multitud de ocasiones. Pero no ocurre lo mismo con el sistema de audio, ya que no en todos los intantes las personas se encuentran hablando.

A continuación se muestran diferentes escenas que se observan a lo largo del experimento. Hay que aclarar que en los siguientes ejemplos el grid que aparece de color azul es el procedente del sistema de visión, y el grid de color rojo procedente del audio.

En las siguientes figuras se observan diferentes ejemplos de un mismo experimento, siendo éste AIT de CLEAR 2007. En el mismo se han empleado cuatro servidores: uno de audio y tres de vídeo. Para calcular el grid de audio se han empleado tres arrays, cada uno de ellos compuesto por cuatro micrófonos. Para calcular el gfrid de vídeo se han utilizado tres cámaras en el experimento. En las imágenes sólo se observan dos de las tres escenas capturadas por las tres cámaras ya que la aplicación sólo dispone de cuatro ventanas de representación. En la ventana de arriba a la izquierda se representa el grid de audio; en las ventanas de arriba a la derecha y abajo a la izquierda aparecen las imágenes capturadas por dos de las cámaras; por último en la ventana de abajo a la derecha se observa el grid final resultando de la fusión audiovisual.

En la Figura 5.15 se muestra un caso en el que aparecen dos personas en la escena, sin embargo el grid de vídeo es nulo ya que una de las dos personas no es capturada a la vez por las dos cámaras, y la otra se encuentra situada por debajo de la altura de 1700 mm, por lo que no se tiene información de ella. En cambio, se encuentra grid procedente del sistema de audio, y esto es debido a que en ese instante la persona que se encuentra de pie en la imagen procedente de la Cámara 1 está hablando.

En la Figura 5.16 no existe ningún ruido o señal de habla, pero sí se realiza el seguimiento de una persona en la escena, por lo que se observa el grid procedente del sistema de visión.

Por último, en la Figura 5.17 se muestra la fusión de ambos grids. La persona de pie en la escena de la Cámara 2 está hablando en ese momento, siendo además la única capturada por el sistema de visión.

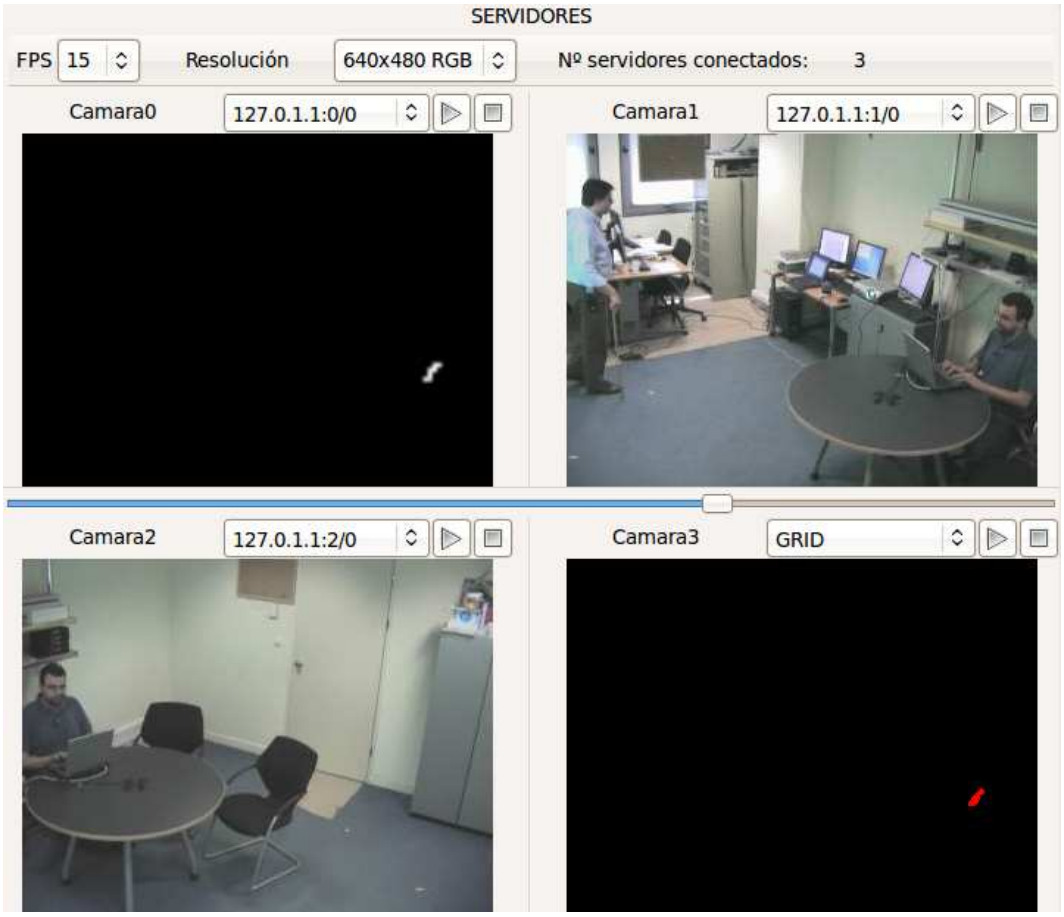


Figura 5.15: Escena de un experimento donde aparece grid procedente del sistema de audio

Figura 5.16: Escena de un experimento donde aparece grid procedente del sistema de visión

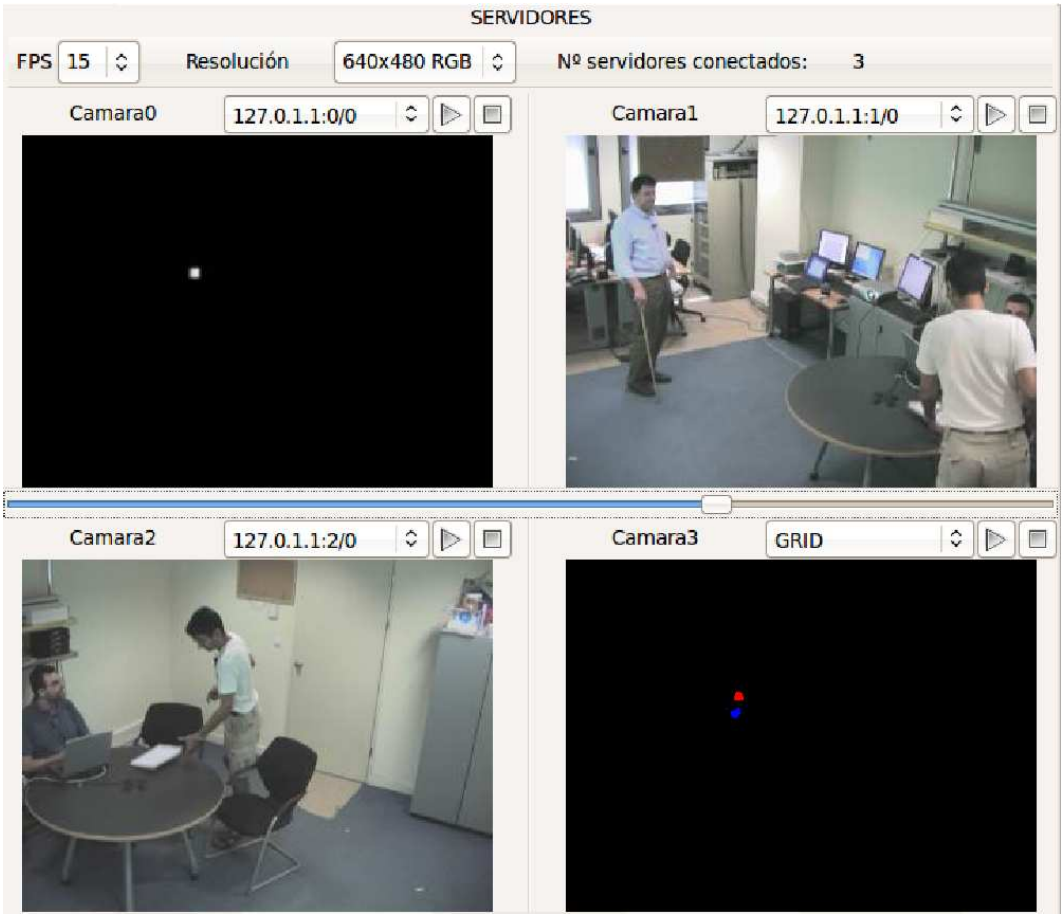


Figura 5.17: Escena de un experimento donde aparece grid procedente del sistema de visión y audio

5.6. Conclusiones

En este capítulo se ha hecho una descripción detallada del diseño e implementación de las mejoras y nuevas técnicas y algoritmos resultantes del trabajo de este proyecto fin de carrera. Dicho trabajo incluye mejoras en los sistemas de localización basados en audio y vídeo, además de una nueva propuesta de estrategia de fusión audiovisual y la mejora del sistema de experimentación disponible

Capítulo 6

Resultados experimentales

6.1. Introducción

A lo largo de este capítulo se exponen los resultados obtenidos relacionados con los sistemas de experimentación empleados en este trabajo. En el caso del sistema de experimentación basado en audio no se ha hecho una evaluación intensiva de la algorítmica y los parámetros de control disponibles (algo que sí se hizo en [8]), sino que se ofrecen como resultados de un sistema base que puede servir de punto de referencia para futuros trabajos de experimentación. En el caso del sistema de localización acústica se muestran los resultados mediante una serie de tablas con los errores cometidos en la estimación, mientras que en el sistema de localización audiovisual se utilizan las propias imágenes capturadas para ilustrar distintas situaciones observadas.

6.2. Resultados del sistema de localización acústica

Para probar el algoritmo SRP se experimenta sobre las bases de datos proporcionadas en el proyecto CHIL (*Computers in the Human Interactive Loop*), siendo éste un proyecto integrado financiado por la Unión Europea bajo el Sexto Programa Marco. Se puede obtener más información acerca de este proyecto y las bases de datos distribuidas en el Capítulo Apéndices en la página 169. A lo largo de este trabajo se han realizado experimentos de CLEAR 2006 y CLEAR 2007 que se muestran a continuación. Además, se han llevado a cabo experimentos con las bases de datos Av-16.3 e HIFI, mostrándose también los resultados obtenidos.

Para comprobar la exactitud de los algoritmos se emplea la evaluación CLEAR (*Classification of Events, Activities and Relationships*), usada en las competiciones. Esta evaluación supone un esfuerzo para caracterizar sistemas que son diseñados con el fin de analizar a las personas, sus identidades, actividades, interacciones y relaciones en escenarios con interacción hombre-hombre. CLEAR nació con el objetivo de aunar proyectos e investigadores en tecnologías relacionadas para establecer una evaluación común internacional.

6.2.1. CLEAR 2006

Para la evaluación CLEAR 2006 se recolectó un set de datos consistente en unas grabaciones audiovisuales de seminarios en distintas salas: AIT, UKA, ITC, UPC e IBM. Todos los datos relativos a estas grabaciones o a otros aspectos como la disposición de los micrófonos y cámaras en las salas o las métricas de evaluación empleadas entre otros se puede encontrar en [74].

Todos los experimentos que se muestran se han realizado con datos de test, no existiendo datos de desarrollo empleados para poner a punto los sistemas.

El método de agrupación de arrays en subarrays se ha empleado en la totalidad de los experimentos mostrados a continuación, ya que se ha demostrado anteriormente que se consiguen mejores resultados.

En todos los experimentos de CLEAR 2006 se han empleado los mismos parámetros de configuración del SRP, y son los siguientes:

```
FRAME_SIZE_SECS=0.500
FFT_SIZE=32768
FRAME_SHIFT_SECS=0.500
WINDOW_TYPE='m'
MAX_DIST=150
```

```
INT_RATE=2
HIGH_FREQ=8000
LOW_FREQ=1000
COARSE2FINE=0
FS=44100
INI_AUDIO_FILE=0
END_AUDIO_FILE=0
DISCARD_FLAG=0
CORR_TYPE=3
INTERPOLATE=0
NUM_MAXS=1
FREQ_SRP_FLAG=0
DISCARD_DEAD_AREAS=0
FILTER_FLAG=0
NOISE_THRESHOLD=24
NOISE_MASKING_FLAG=0
ROUND_FLAG=1
DIST_WEIGHTING_FLAG=0
FIXED_THRESHOLD_FLAG=0
NOISE_SIZE_SECS=0.05
CHANNELS=´´all"
```

6.2.1.1. University of Karlsruhe UKA

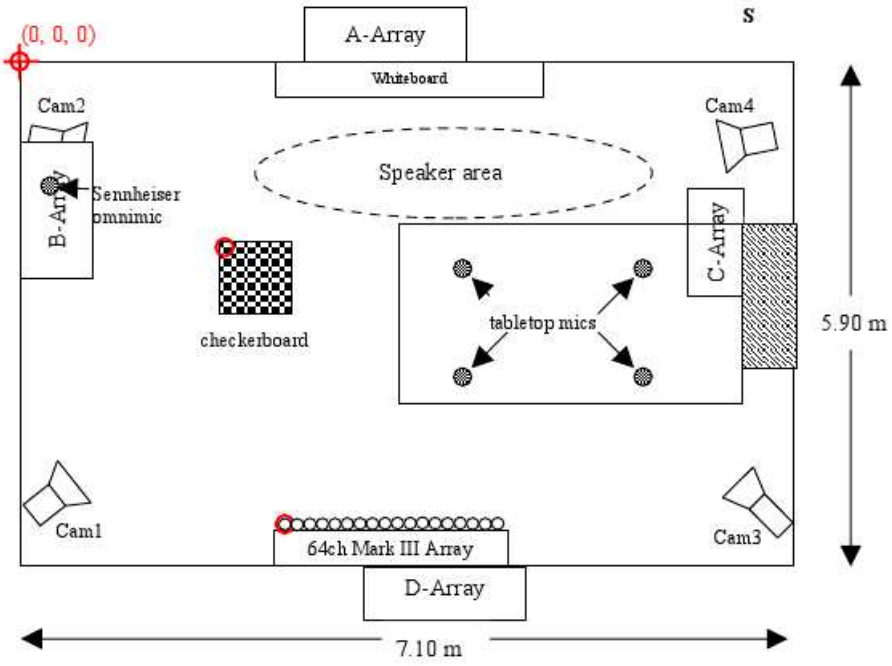


Figura 6.1: Sala de grabación de UKA

Se observa en la Figura 6.1 que en esta habitación existen 4 arrays de 4 micrófonos cada uno y un array de 64 canales Mark III. Para la realización del experimento se emplean únicamente los 4 arrays en forma de T (4 micrófonos).

En la Tabla 6.1 se observan los resultados obtenidos.

Tabla 6.1: Resultados obtenidos con el set de test de UKA

	Results
Pcor Rel. error reduction	$57{,}0 \pm 1{,}4\,\%$
Bias fine (x:y:z) [mm]	$20 : -42 : -75$
Bias fine+gross (x,y,z) [mm]	$735 : -93 : -258$
AEE fine [mm] = MOTP Rel. AEE reduction	210
Fine+gross [mm] Rel. BIAS f+g reduction	1201
Loc. frames	5035
Ref. duration (s)	6287,0

6.2.1.2. Istituto Trentino di Cultura ITC

Para este experimento se emplean los 7 arrays de micrófonos presentes en la sala y que se observan en la Figura 6.2, numerados de T0 a T6.

En la Tabla 6.2 se muestran los resultados conseguidos en este experimento.

Tabla 6.2: Resultados obtenidos con el set de test de ITC

	Results
Pcor Rel. error reduction	$84{,}0 \pm 3{,}3\,\%$
Bias fine (x:y:z) [mm]	$45 : 27 : -41$
Bias fine+gross (x,y,z) [mm]	$67 : 439 : -134$
AEE fine [mm] = MOTP Rel. AEE reduction	130
Fine+gross [mm] Rel. BIAS f+g reduction	632
Loc. frames	479
Ref. duration (s)	596,0

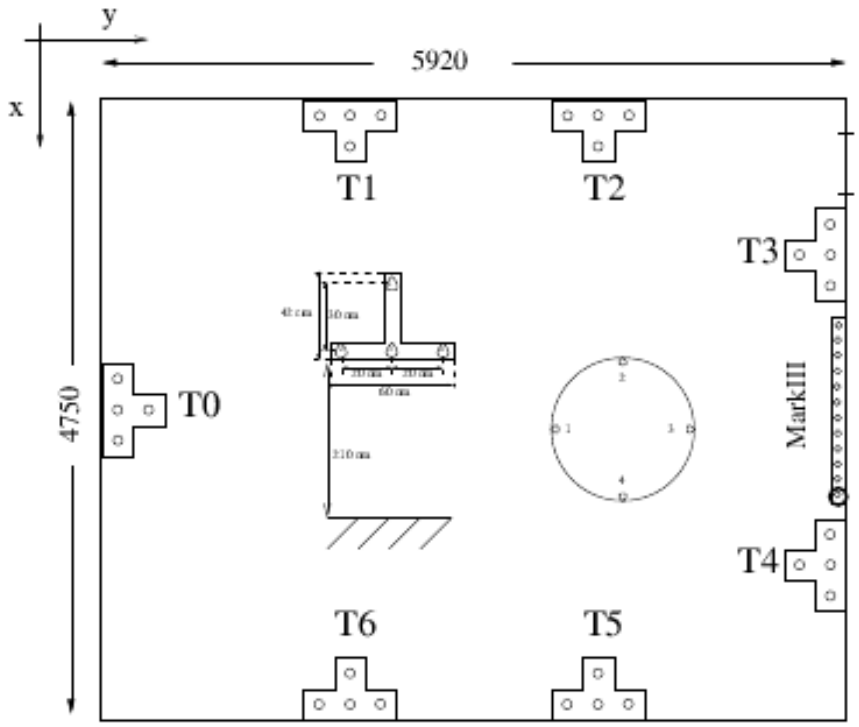


Figura 6.2: Sala de grabación de ITC

6.2.1.3. Research and Education Society in Information Technololgy AIT RESIT

En esta sala existen 3 arrays de 4 micrófonos en forma de T (A, C y D en la figura 6.3) que han sido utilizados para la realización del experimento cuyos resultados se muestran a continuación. Además, aparece un array de 64 canales que no ha sido empleado para la experimentación.

Las tasa de error obtenidas en este experimento se pueden ver en la Tabla 6.3.

6.2.1.4. Universitat Politècnica de Catalunya UPC

Para este experimento se han utilizado los 3 arrays de 4 micrófonos que se observan en la Figura 6.4. El array de 64 canales Mark III no se ha empleado, al igual que en el resto de experimentos.

Los resultados obtenidos se muestran en la Tabla 6.4.

6.2.1.5. IBM

En esta sala se encuentran 4 arrays de micrófonos en forma de T con 4 micrófonos cada uno más 2 arrays de 64 canales, no empleados en este experimento. Esto se puede ver en la Figura 6.5.

En la Tabla 6.5 se observan las tasas de error conseguidas mediante la realización de este experimento.

En la Tabla 6.6 se muestra un resumen de los resultados obtenidos en cada uno de los

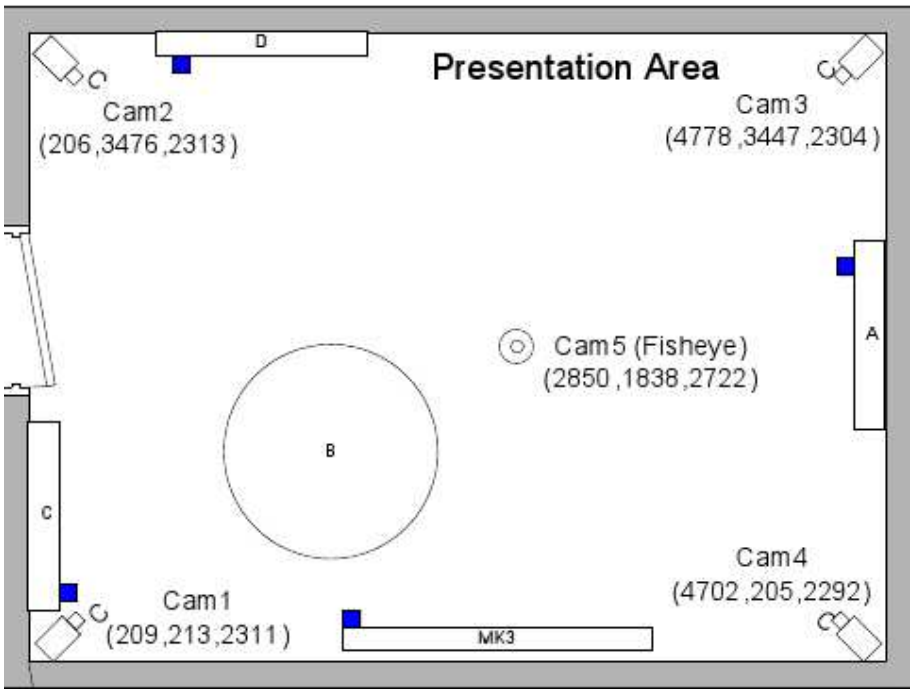


Figura 6.3: Sala de grabación de AIT RESIT

Tabla 6.3: Resultados obtenidos con el set de test de AIT

	Results
Pcor Rel. error reduction	$47{,}0 \pm 3{,}1\,\%$
Bias fine (x:y:z) [mm]	$-27 : -77 : -40$
Bias fine+gross (x,y,z) [mm]	$17 : -402 : -118$
AEE fine [mm] = MOTP Rel. AEE reduction	266
Fine+gross [mm] Rel. BIAS f+g reduction	1006
Loc. frames	995
Ref. duration (s)	1143,0

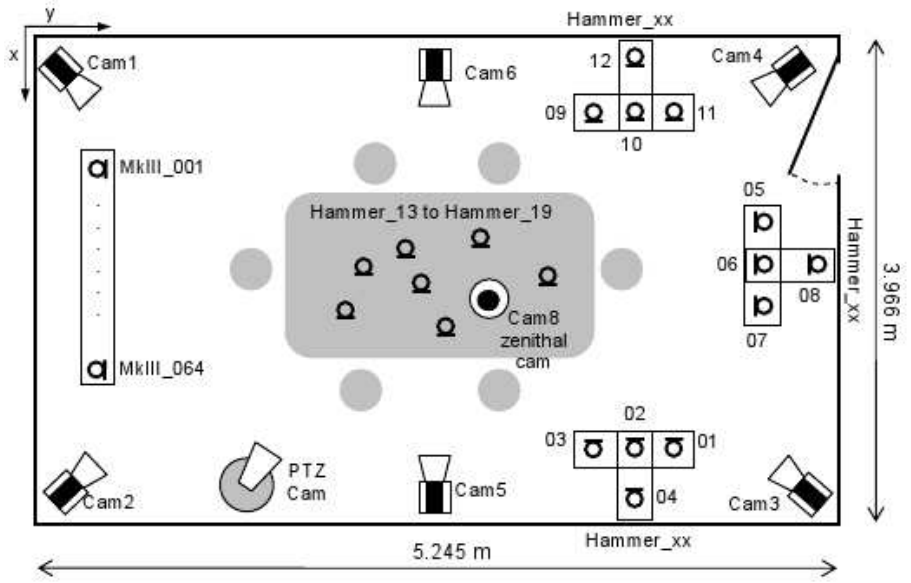


Figura 6.4: Sala de grabación de UPC

Tabla 6.4: Resultados obtenidos con el set de test de UPC

	Results
Pcor Rel. error reduction	$20{,}0 \pm 2{,}5\,\%$
Bias fine (x:y:z) [mm]	$-59 : 112 : 52$
Bias fine+gross (x,y,z) [mm]	$-141 : 255 : 39$
AEE fine [mm] = MOTP Rel. AEE reduction	344
Fine+gross [mm] Rel. BIAS f+g reduction	1188
Loc. frames	977
Ref. duration (s)	1180,0

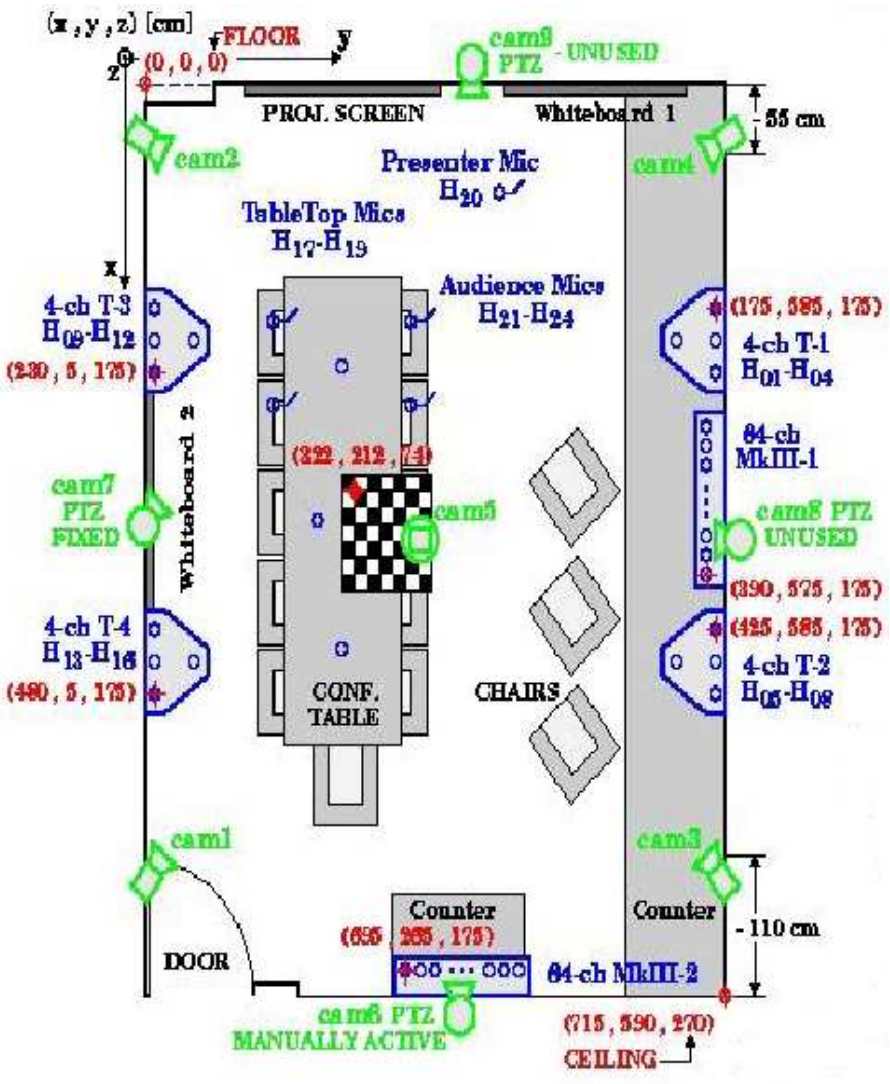


Figura 6.5: Sala de grabación de IBM

Tabla 6.5: Resultados obtenidos con el set de test de IBM

	Results
Pcor Rel. error reduction	$67{,}0 \pm 2{,}9\,\%$
Bias fine (x:y:z) [mm]	$91 : -69 : -38$
Bias fine+gross (x,y,z) [mm]	$472 : -141 : -14$
AEE fine [mm] = MOTP Rel. AEE reduction	228
Fine+gross [mm] Rel. BIAS f+g reduction	883
Loc. frames	1025
Ref. duration (s)	1194,0

experimentos.

En [75] se muestran los resultados conseguidos por los diferentes institutos, universidades y centros tecnológicos en los experimentos que han llevado a cabo, mostrando sus tasas de error. Los mejores resultados han sido obtenidos por UKA con un valor de MOTP de 186 mm, mientras que AIT presenta la tasa más elevada de error, con un MOTP de 226 mm. Observando estos valores, se puede destacar los resultados conseguidos en nuestro experimento ITC, donde el valor de MOTP es de 130 mm, ver Tabla 6.2.

6.2.2. CLEAR 2007

Los datos recolectados por CHIL en el año 2007 emplearon las técnicas de evaluación correspondientes a CLEAR. Se muestran posteriormente los resultados obtenidos por el algoritmo SRP en cada una de las salas, siendo éstas las mismas que en el caso de CLEAR 2006.

Todos los experimentos mostrados se han realizado empleando la técnica ya explicada de agrupación de micrófonos, ya que como se comentó los resultados son considerablemente mejores.

Además, para cada seminario de cada base de datos se proporcionan datos de *test* y de *desarrollo*. Los datos de desarrollo son entregados para poner a punto los sistemas, y así posteriormenter evaluar sus algoritmos empleando los datos de test.

En [70] se puede encontrar toda la información detallada sobre los seminarios, las métricas empleadas, características físicas de las salas, etc.

Se han establecido los mismos parámetros de configuración de SRP para todos los experimentos de CLEAR 2007, los cuales se muestran a continuación:

```
FRAME_SIZE_SECS=0.500
FFT_SIZE=32768
FRAME_SHIFT_SECS=0.500
WINDOW_TYPE='m'
MAX_DIST=150
INT_RATE=2
HIGH_FREQ=8000
LOW_FREQ=1000
COARSE2FINE=0
FS=44100
INI_AUDIO_FILE=0
END_AUDIO_FILE=0
DISCARD_FLAG=0
CORR_TYPE=3
INTERPOLATE=0
NUM_MAXS=1
FREQ_SRP_FLAG=0
DISCARD_DEAD_AREAS=0
FILTER_FLAG=0
NOISE_THRESHOLD=24
NOISE_MASKING_FLAG=0
ROUND_FLAG=1
DIST_WEIGHTING_FLAG=0
FIXED_THRESHOLD_FLAG=0
NOISE_SIZE_SECS=0.05
CHANNELS=´´all"
```

	UKA	ITC	AIT	UPC	IBM
Pcor	$57{,}0 \pm 1{,}4\,\%$	$84{,}0 \pm 3{,}3\,\%$	$47{,}0 \pm 3{,}1\,\%$	$20{,}0 \pm 2{,}5\,\%$	$67{,}0 \pm 2{,}9\,\%$
Bias fine (x:y:z) [mm]	$20 : -42 : -75$	$45 : 27 : -41$	$-27 : -77 : -40$	$-59 : 112 : 52$	$91 : -69 : -38$
Bias fine+gross (x,y,z) [mm]	$735 : -93 : -258$	$67 : 439 : -134$	$17 : -402 : -118$	$-141 : 255 : 39$	$474 : -141 : -14$
AEE fine [mm] = MOTP	210	130	266	344	228
Fine+gross [mm]	1201	632	1006	1188	884
Loc. frames	5035	22	995	977	1023
Ref. duration (s)	6287,0	596,0	1143,0	1180,0	1194,0

Tabla 6.6: Resultados TEST CLEAR 2006

6.2.2.1. University of Karlsruhe UKA

En esta sala existen 4 arrays de micrófonos de 4 micrófonos cada uno (A, B, C y D en la Figura 6.1) y un array de 64 canales tipo Mark III. Para el experimento se han empleado los 4 arrays únicamente.

En las Tablas 6.7 y 6.8 se muestran los resultados obtenidos en este experimento.

Tabla 6.7: Resultados obtenidos con el set de test de UKA

	Results
Pcor Rel. error reduction	$67{,}0 \pm 4{,}2\,\%$
Bias fine (x:y:z) [mm]	$-13 : -31 : -46$
Bias fine+gross (x,y,z) [mm]	$57 : -92 : -82$
AEE fine [mm] = MOTP Rel. AEE reduction	172
Fine+gross [mm] Rel. BIAS f+g reduction	613
Loc. frames	476
Ref. duration (s)	2398,0

Tabla 6.8: Resultados obtenidos con el set de desarrollo de UKA

	Results
Pcor Rel. error reduction	$71{,}0 \pm 2{,}8\,\%$
Bias fine (x:y:z) [mm]	$62 : -19 : -44$
Bias fine+gross (x,y,z) [mm]	$325 : 27 : -76$
AEE fine [mm] = MOTP Rel. AEE reduction	143
Fine+gross [mm] Rel. BIAS f+g reduction	631
Loc. frames	990
Ref. duration (s)	1200,0

6.2.2.2. Istituto Trentino di Cultura ITC

Para realizar los experimentos de esta sala se han tenido en cuenta los 7 arrays de micrófonos existentes (T0,...T6 en la Figura 6.2).Además hay también un array de 64 canales Mark III en la parte derecha de la sala.

En las Tablas 6.9 y 6.10 se pueden comprobar las tasas de error obtenidas en el experimento.

Tabla 6.9: Resultados obtenidos con el set de test de ITC

	Results
Pcor Rel. error reduction	$55{,}0 \pm 2{,}3\,\%$
Bias fine (x:y:z) [mm]	$-4 : -45 : -57$
Bias fine+gross (x,y,z) [mm]	$128 : -161 : -138$
AEE fine [mm] = MOTP Rel. AEE reduction	151
Fine+gross [mm] Rel. BIAS f+g reduction	643
Loc. frames	1822
Ref. duration (s)	2470,0

Tabla 6.10: Resultados obtenidos con el set de desarrollo de ITC

	Results
Pcor Rel. error reduction	$62{,}0 \pm 3{,}0\,\%$
Bias fine (x:y:z) [mm]	$-110 : 20 : -6$
Bias fine+gross (x,y,z) [mm]	$-51 : 415 : 25$
AEE fine [mm] = MOTP Rel. AEE reduction	175
Fine+gross [mm] Rel. BIAS f+g reduction	726
Loc. frames	993
Ref. duration (s)	1208,0

6.2.2.3. Research and Education Society in Information Technololgy AIT RESIT

En la sala AIT aparecen 3 arrays de 4 micrófonos cada uno y un array de 64 canales Mark III. Para los experimentos sólo se han empleado los 3 arrays de 4 micrófonos.

Los resultados conseguidos tras las realización de este experimento se observan en las Tablas 6.11 y 6.12.

Tabla 6.11: Resultados obtenidos con el set de test de AIT

	Results
Pcor Rel. error reduction	$65{,}0 \pm 2{,}6\,\%$
Bias fine (x:y:z) [mm]	$-5 : -30 : -107$
Bias fine+gross (x,y,z) [mm]	$-9 : -81 : -163$
AEE fine [mm] = MOTP Rel. AEE reduction	226
Fine+gross [mm] Rel. BIAS f+g reduction	620
Loc. frames	1273
Ref. duration (s)	2364,0

Tabla 6.12: Resultados obtenidos con el set de desarrollo de AIT

	Results
Pcor Rel. error reduction	$85{,}0 \pm 2{,}5\,\%$
Bias fine (x:y:z) [mm]	$-109 : -28 : -127$
Bias fine+gross (x,y,z) [mm]	$-216 : -69 : -154$
AEE fine [mm] = MOTP Rel. AEE reduction	220
Fine+gross [mm] Rel. BIAS f+g reduction	378
Loc. frames	797
Ref. duration (s)	1200,0

6.2.2.4. Universitat Politècnica de Catalunya UPC

Al igual que en las demás salas existe un array de 64 canales Mark III, además aparecen 3 arrays de 4 micrófonos los cuales se han empleado en los experimentos para la obtención de los

resultados mostrados a continuación en las Tablas 6.13 y 6.14.

Se pueden comprobar las tasas de error obtenidas en este experimento en las Tablas 6.13 y 6.14.

Tabla 6.13: Resultados obtenidos con el set de test de UPC

	Results
Pcor Rel. error reduction	68,0 ± 2,1 %
Bias fine (x:y:z) [mm]	36 : 50 : −34
Bias fine+gross (x,y,z) [mm]	75 : −80 : −139
AEE fine [mm] = MOTP Rel. AEE reduction	159
Fine+gross [mm] Rel. BIAS f+g reduction	628
Loc. frames	1819
Ref. duration (s)	2481,0

Tabla 6.14: Resultados obtenidos con el set de desarrollo de UPC

	Results
Pcor Rel. error reduction	65,0 ± 2,8 %
Bias fine (x:y:z) [mm]	56 : 40 : −56
Bias fine+gross (x,y,z) [mm]	58 : −165 : −219
AEE fine [mm] = MOTP Rel. AEE reduction	157
Fine+gross [mm] Rel. BIAS f+g reduction	614
Loc. frames	1084
Ref. duration (s)	1375,0

6.2.2.5. IBM

En la sala IBM existen 4 arrays de 4 micrófonos cada uno más 2 arrays de 64 canales Mark III. Para la ejecución de los experimentos mostrados sólo se han tenido en cuenta los 4 arrays de micrófonos.

Los resultados conseguidos se muestran en las Tablas 6.15 y 6.16.

Tabla 6.15: Resultados obtenidos con el set de test de IBM

	Results
Pcor Rel. error reduction	$63{,}0 \pm 2{,}2\,\%$
Bias fine (x:y:z) [mm]	$96 : -46 : -62$
Bias fine+gross (x,y,z) [mm]	$470 : -200 : -100$
AEE fine [mm] = MOTP Rel. AEE reduction	236
Fine+gross [mm] Rel. BIAS f+g reduction	904
Loc. frames	1891
Ref. duration (s)	2427,0

Tabla 6.16: Resultados obtenidos con el set de desarrollo de IBM

	Results
Pcor Rel. error reduction	$79{,}0 \pm 2{,}5\,\%$
Bias fine (x:y:z) [mm]	$142 : -75 : -67$
Bias fine+gross (x,y,z) [mm]	$350 : -212 : -81$
AEE fine [mm] = MOTP Rel. AEE reduction	251
Fine+gross [mm] Rel. BIAS f+g reduction	619
Loc. frames	989
Ref. duration (s)	1200,0

Un resumen de los resultados obtenidos en cada experimento se puede observar en las Tablas 6.17 y 6.18.

	UKA	ITC	AIT	UPC	IBM
Pcor	$67{,}0 \pm 4{,}2\,\%$	$55{,}0 \pm 2{,}3\,\%$	$65{,}0 \pm 2{,}6\,\%$	$68{,}0 \pm 2{,}1\,\%$	$63{,}0 \pm 2{,}2\,\%$
Bias fine (x:y:z) [mm]	$-13 : -31 : -46$	$-4 : -45 : -57$	$-5 : -30 : -107$	$36 : 50 : -34$	$96 : -46 : -62$
Bias fine+gross (x,y,z) [mm]	$57 : -92 : -82$	$128 : -161 : -138$	$-9 : -81 : -163$	$75 : -80 : -139$	$470 : -200 : -100$
AEE fine [mm] = MOTP	172	151	226	159	236
Fine+gross [mm]	613	643	620	628	904
Loc. frames	476	1822	1273	1819	1891
Ref. duration (s)	2398,0	2470,0	2364,0	2481,0	2427,0

Tabla 6.17: Resultados TEST CLEAR 2007

En [76] se muestran los resultados obtenidos por diferentes sistemas desarrollados en distintas universidades o institutos para la evaluación CLEAR 2007, siendo UKA la que mejores puntaciones ha logrado con una tasa de Pcor del 54,63 % y una precisión de 140 mm (MOTP). Como se puede comprobar en la Tabla 6.12 en este trabajo se ha alcanzado una tasa Pcor del 85 % muy superior a la obtenida por UKA; sin embargo la precisión obtenida es peor, siendo ésta de 220 mm (MOTP).

6.2.3. Av-16.3

También se ha llevado a cabo un experimento con la base de datos Av-16.3, empleando las grabaciones de locutores estáticos donde dichos locutores únicamente emiten sonido desde unas posiciones definidas previamente. En dichos experimentos los parámetros empleados han sido los que se muestran a continuación:

```
FRAME_SIZE_SECS=0.500
FFT_SIZE=8192
FRAME_SHIFT_SECS=0.04
WINDOW_TYPE='m'
MAX_DIST=150
INT_RATE=2
HIGH_FREQ=8000
LOW_FREQ=4000
COARSE2FINE=0
FS=16000
INI_AUDIO_FILE=0
END_AUDIO_FILE=0
DISCARD_FLAG=0
CORR_TYPE=3
INTERPOLATE=0
NUM_MAXS=3
FREQ_SRP_FLAG=0
DISCARD_DEAD_AREAS=0
FILTER_FLAG=1
NOISE_THRESHOLD=24
NOISE_MASKING_FLAG=0
ROUND_FLAG=1
DIST_WEIGHTING_FLAG=0
FIXED_THRESHOLD_FLAG=0
NOISE_SIZE_SECS=0.05
CHANNELS=´´all"
```

6.2.3.1. Experimento Av-16.3 con subarrays

Se ha realizado el experimento utilizando la técnica de agrupación de micrófonos en subarrays. Los resultados obtenidos se muestran en la Tabla 6.19.

	UKA	ITC	AIT	UPC	IBM
Pcor	$71{,}0 \pm 2{,}8\,\%$	$62{,}0 \pm 3{,}0\,\%$	$85{,}0 \pm 2{,}5\,\%$	$65{,}0 \pm 2{,}8\,\%$	$79{,}0 \pm 2{,}5\,\%$
Bias fine (x:y:z) [mm]	$62 : -19 : -44$	$-110 : 20 : -6$	$-109 : -28 : -127$	$56 : 40 : -56$	$142 : -75 : -67$
Bias fine+gross (x,y,z) [mm]	$325 : 27 : -76$	$-51 : 415 : 25$	$-216 : -69 : -154$	$58 : -165 : -219$	$350 : -212 : -81$
AEE fine [mm] = MOTP	143	175	220	157	251
Fine+gross [mm]	631	726	378	614	619
Loc. frames	990	993	797	1084	989
Ref. duration (s)	1200,0	1208,0	1200,0	1375,0	1200,0

Tabla 6.18: Resultados DEV CLEAR 2007

Tabla 6.19: Resultados obtenidos empleando subarrays en AV-16.3

	Results
Pcor Rel. error reduction	$95{,}0 \pm 0{,}5\,\%$
Bias fine (x:y:z) [mm]	$13 : -20 : -36$
Bias fine+gross (x,y,z) [mm]	$-7 : -52 : -42$
AEE fine [mm] = MOTP Rel. AEE reduction	137
Fine+gross [mm] Rel. BIAS f+g reduction	211
Loc. frames	7295
Ref. duration (s)	600,0

6.2.4. HIFI

Se han llevado a cabo experimentos con la base de datos HIFI. Los parámetros utilizados para llevar a cabo este experimento son los siguientes:

```
FRAME_SIZE_SECS=0.32
FFT_SIZE=16384
FRAME_SHIFT_SECS=0.04
WINDOW_TYPE='m'
MAX_DIST=150
INT_RATE=2
HIGH_FREQ=16000
LOW_FREQ=2000
COARSE2FINE=0
FS=48000
INI_AUDIO_FILE=0
END_AUDIO_FILE=0
DISCARD_FLAG=0
CORR_TYPE=3
INTERPOLATE=0
NUM_MAXS=3
FREQ_SRP_FLAG=0
DISCARD_DEAD_AREAS=0
FILTER_FLAG=1
NOISE_THRESHOLD=24
NOISE_MASKING_FLAG=0
ROUND_FLAG=1
DIST_WEIGHTING_FLAG=0
FIXED_THRESHOLD_FLAG=0
NOISE_SIZE_SECS=0.05
```

```
CHANNELS=´´all"
```

6.2.4.1. Experimento HIFI completo

En primer lugar se ha realizado un experimento considerando todos los los locutores y posiciones tenidas en cuenta en la grabación a la hora de calcular el error entre la posición estimada por el sistema y la posición real de cada uno de ellos. Los resultados obtenidos se observan en la Tabla 6.20.

Tabla 6.20: Resultados obtenidos con todos los locutores en HIFI

	Results
Pcor Rel. error reduction	$88{,}0 \pm 0{,}7\,\%$
Bias fine (x:y:z) [mm]	$-1 : -21 : -50$
Bias fine+gross (x,y,z) [mm]	$3 : 21 : -56$
AEE fine [mm] = MOTP Rel. AEE reduction	195
Fine+gross [mm] Rel. BIAS f+g reduction	375
Loc. frames	9390
Ref. duration (s)	367,0

6.2.4.2. Experimento HIFI en función del locutor

En las Tablas 6.21 y 6.22 se muestran los resultados individuales para cada uno de los locutores, es decir, el error cometido en la estimación de la posición de cada uno de ellos.

	AGR	CCG	CRL	FFM	IFG	JMG
Pcor	$88{,}0 \pm 2{,}3\,\%$	$91{,}0 \pm 2{,}1\,\%$	$89{,}0 \pm 2{,}3\,\%$	$87{,}0 \pm 2{,}3\,\%$	$84{,}0 \pm 3{,}1\,\%$	$84{,}0 \pm 1{,}9\,\%$
Bias fine (x:y:z) [mm]	$-0 : -42 : -119$	$27 : -29 : -129$	$16 : 27 : 17$	$31 : -56 : -41$	$48 : -34 : 70$	$-34 : -10 : -125$
Bias fine+gross (x,y,z) [mm]	$-28 : 2 : -121$	$14 : 19 : -125$	$5 : 17 : 13$	$38 : 10 : -55$	$31 : 31 : 62$	$-32 : 12 : -135$
AEE fine [mm] = MOTP	170	213	179	195	188	187
Fine+gross [mm]	354	304	330	369	442	445
Loc. frames	752	682	733	806	524	1409
Ref. duration (s)	29,0	27,0	29,0	31,0	20,0	55,0

Tabla 6.21: Resultados obtenidos para cada locutor en HIFI

Como se comprueba, en la estimación de la posición de la mayor parte de los locutores se comete un error caracterizado por una Pcor de un 90 %. Únicamente en dos casos, locutores IFG y JMG, este valor se encuentra por debajo del 85 %.

6.2.4.3. Experimento HIFI en función de la posición

A la hora de realizar las grabaciones en la sala HIFI, cada uno de los locutores se colocaba en una serie de posiciones marcadas, concretamente 5, únicas posiciones desde donde dichos locutores han hablado.

En la Tabla 6.23 se muestran las tasas de error conseguidas para cada una de dichas posiciones de forma individual.

Los resultados obtenidos para cada una de las posiciones son semejantes, con un valor de Pcor de 90 % aproximadamente, con excepción de la posiciones 2 y 4 cuya PCcor es menor, alrededor de un 85 %.

6.2.4.4. Experimento HIFI en función de los micrófonos empleados

Los experimentos mostrados hasta ahora de la base de datos HIFI se han desarrollado empleando únicamente un array de 4 micrófonos. Además de este array en la sala existe otro de 3 canales. Se han realizado experimentos empleando cada uno de ellos por separado y también utilizando ambos para calcular las estimaciones.

En la Tabla 6.24 se muestran los errores cometidos en cada uno de los casos.

Se comprueba que los mejores resultados son los obtenidos con el array de 4 micrófonos, seguidos por la configuración de 3 canales, y como peor resultados el conseguido al emplear para la localización ambos arrays.

6.3. Resultados del sistema de experimentación basado en fusión audiovisual

La comprobación del funcionamiento del sistema de fusión audiovisual se puede realizar:

- De manera visual, comprobando en cada frame el error cometido.
- Mediante tablas con el mismo formato de CLEAR como las mostradas en la sección anterior.

En este trabajo sólo se emplea la primera de ellas, aunque se propone como línea futura realizar la evaluación CLEAR de los resultados.

6.3.1. Evaluación de los resultados obtenidos

Con el fin de evaluar la solución propuesta para la estimación de la posición de hablantes mediante técnicas audiovisuales se ha realizado un experimento de la sala AIT, del set de desarrollo o DEV, concretamente el seminario AIT_20060728. En este seminario existen 5 cámaras, como se puede observar en la Figura 6.3, 4 de ellas posicionadas en las esquinas de las paredes

	LFD	NSI	RAP	RBC	RRB	XMN
Pcor	$90{,}0 \pm 2{,}1\,\%$	$90{,}0 \pm 2{,}1\,\%$	$90{,}0 \pm 1{,}7\,\%$	$90{,}0 \pm 2{,}2\,\%$	$87{,}0 \pm 3{,}1\,\%$	$90{,}0 \pm 2{,}5\,\%$
Bias fine (x:y:z) [mm]	$17 : -53 : -148$	$27 : -20 : 11$	$-35 : -7 : 0$	$-33 : -42 : -54$	$-59 : 2 : 94$	$22 : 10 : -43$
Bias fine+gross (x,y,z) [mm]	$50 : 31 : -151$	$29 : 15 : 5$	$-44 : 28 : -3$	$-5 : 11 : -62$	$-32 : -6 : 87$	$76 : 93 : -56$
AEE fine [mm] = MOTP	197	159	185	231	227	253
Fine+gross [mm]	373	321	322	411	405	427
Loc. frames	811	757	1154	735	465	562
Ref. duration (s)	32,0	30,0	45,0	29,0	18,0	22,0

Tabla 6.22: Resultados obtenidos para cada locutor en HIFI

	P1	P2	P3	P4	P5
Pcor	$90{,}0 \pm 1{,}3\,\%$	$85{,}0 \pm 1{,}7\,\%$	$90{,}0 \pm 1{,}3\,\%$	$86{,}0 \pm 1{,}5\,\%$	$89{,}0 \pm 1{,}5\,\%$
Bias fine (x:y:z) [mm]	$-110 : 13 : -43$	$-33 : 13 : -46$	$-7 : -29 : -158$	$69 : -101 : -33$	$77 : 3 : 40$
Bias fine+gross (x,y,z) [mm]	$-261 : 142 : -48$	$-119 : 72 : -52$	$19 : -204 : -151$	$141 : -16 : -45$	$243 : 127 : 23$
AEE fine [mm] = MOTP	210	192	178	197	199
Fine+gross [mm]	415	339	358	350	412
Loc. frames	1929	1761	1970	1945	1785
Ref. duration (s)	75,0	69,0	77,0	76,0	70,0

Tabla 6.23: Resultados obtenidos para cada posición en HIFI

Tabla 6.24: Resultados obtenidos con distintas configuraciones de micrófonos en HIFI

	4mic	3mic	4+3mic
Pcor	$88{,}0 \pm 0{,}7\,\%$	$14{,}0 \pm 0{,}7\,\%$	$83{,}0 \pm 0{,}8\,\%$
Bias fine (x:y:z) [mm]	$-1 : -21 : -50$	$74 : 122 : -101$	$6 : 8 : -34$
Bias fine+gross (x,y,z) [mm]	$3 : 21 : -56$	$-181 : 39 : -134$	$-9 : 53 : -40$
AEE fine [mm] = MOTP	195	305	218
Fine+gross [mm]	375	1629	451
Loc. frames	9390	9390	9390
Ref. duration (s)	367,0	367,0	367,0

de la habitación y la restante en el techo. Además, hay 3 arrays de 4 micrófonos y un array de 64 canales Mark III.

Para la realización del experimento se han empleado únicamente 3 de las 4 cámaras posicionadas en las esquinas y los 3 arrays de 4 micrófonos. Se ha utilizado una matriz de homografía calculada para una altura de 1700 mm, ya que se considera la altura media de una persona.

Los parámetros empleados en el experimento, cuya explicación se puede encontrar en el Manual de Usuario de la página 139, son los siguientes:

- Servidor
  ```
  FRAMES_BACKGROUND=4
  FRAMES_BACKGROUND_AUDIO=87
  OFFSET_BACKGROUND=0
  AUDIO_CALIB=0
  SIMULACION=1
  READREFERENCE=1
  FPS=30
  FPS_AUDIO=10
  SEMINAR=.ªIT_20060728"
  NUM_GRAY_SCALE=256
  ```

- Cliente
  ```
  GRIDSIZE_X=16
  GRIDSIZE_Y=16
  GRIDSTEP=9
  REFRESCO_ZONAS_DIBUJO=33
  ```

Para la correcta evaluación de los experimentos, son suministrados los datos de *ground truth*; es decir, las posiciones reales donde se encuentran las personas que hablan (sistema de experimentación basado en audio) o que simplemente se encuentran en la sala (sistema de experimentación basado en vídeo).

En la aplicación cliente-servidor se representan en las imágenes estos *ground truths*, de manera

que en un instante determinado se observa en una imagen si se debería estar detectando a una persona mediante audio o visión y el resultado final obtenido. En la Figura 6.16 se muestran dos imágenes de la misma escena, donde aparece en color rojo un triángulo indicando la posición real de la persona etiquetada en los *ground truths*.

(a) (b)

Figura 6.6: Etiquetado del *ground truth* en una escena

De la misma manera se indica la posición real de aquellas personas que se encuentren hablando en ese momento en la imagen formada a partir de la información acústica, como se comprueba en la Figura 6.7. En dicha figura aparecen unos triángulos de color verde que indican la posición de los arrays de micrófonos en la sala, y un triángulo de color rojo marcando la posición de la persona que está hablando.

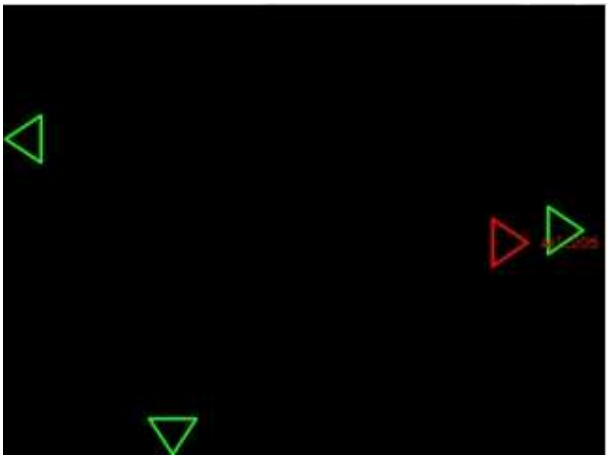


Figura 6.7: Etiquetado del *ground truth* de una imagen con información acústica

En la Figura 6.8 se muestra un ejemplo donde aparece marcada la posición real de la persona que está hablando (triángulo rojo en la imagen de arriba a la izquierda) y también la posición de las personas presentes en la sala (triángulos rojos en las imágenes de arriba a la derecha y abajo izquierda). La imagen de abajo a la derecha corresponde al grid final, procedente de la fusión audiovisual.

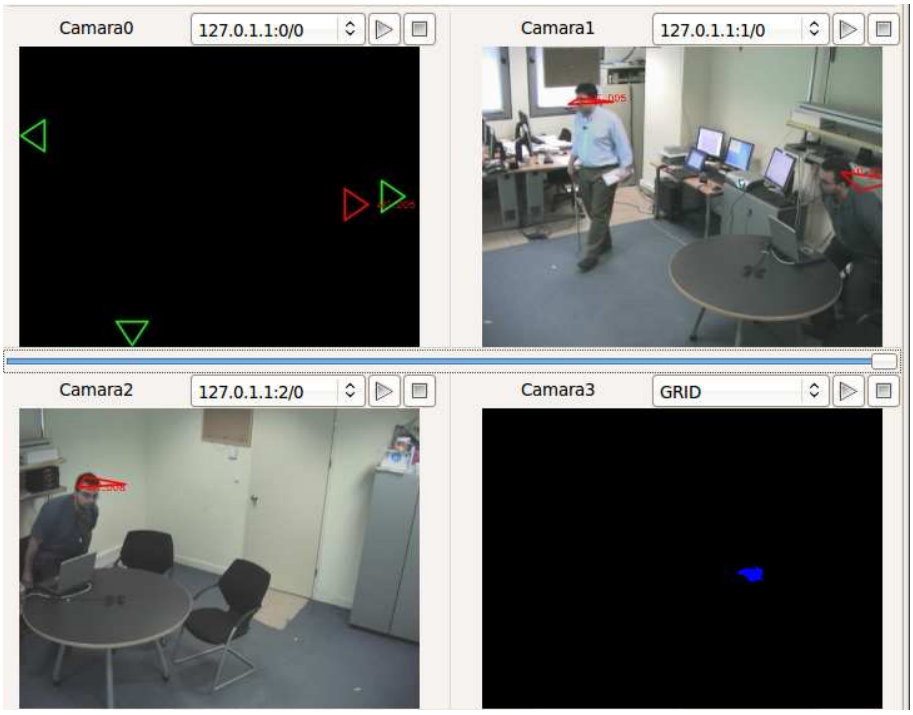


Figura 6.8: Etiquetado del *ground truth* tanto de información acústica como visual

6.3.1.1. Distintas situaciones observadas

En las siguientes figuras se muestran diversas situaciones observadas al realizar el experimento y que son interesantes de comentar dada la situación que se produce.

Antes de nada se explica el significado de cada una de las 4 imágenes mostradas en cada figura:

- Imagen de arriba a la izquierda: Imagen obtenida mediante información acústica. Aparecen una serie de triángulos verdes que representan los arrays de micrófonos; el triángulo rojo marca la posición etiquetada de la persona que se encuentre hablando en ese instante. En caso de que se muestre un grid de audio, es decir, que el sistema considere que en ciertas posiciones existe una persona hablando, aparecen una serie de puntos blancos en las posiciones estimadas formando una mancha del mismo color.
- Imagen de arriba a la derecha y abajo a la izquierda: Imágenes obtenidas de las cámaras de la sala. Aparecen en color rojo los triángulos marcando la posición de las personas basándose en información visual, es decir, de aquellas personas que son visibles desde cada una de las cámaras.
- Imagen de abajo a la derecha: Imagen correspondiente al grid total, fusión del grid de audio y del grid de visión. Cuando el Filtro de Partículas valide dicho grid se pinta un círculo de distintos colores que se representa en todas las imágenes.

En la Figura 6.9 se comprueba que aparece un triángulo rojo en la escena procedente de información acústica (arriba izquierda), por lo tanto existe un etiquetado en ese instante. Como se puede observar, el grid de audio obtenido es exacto ya que coincide perfectamente con la posición real de la persona que se encuentra hablando. En la imagen de abajo a la derecha aparece el Filtro de Partículas validando correctamente el grid.

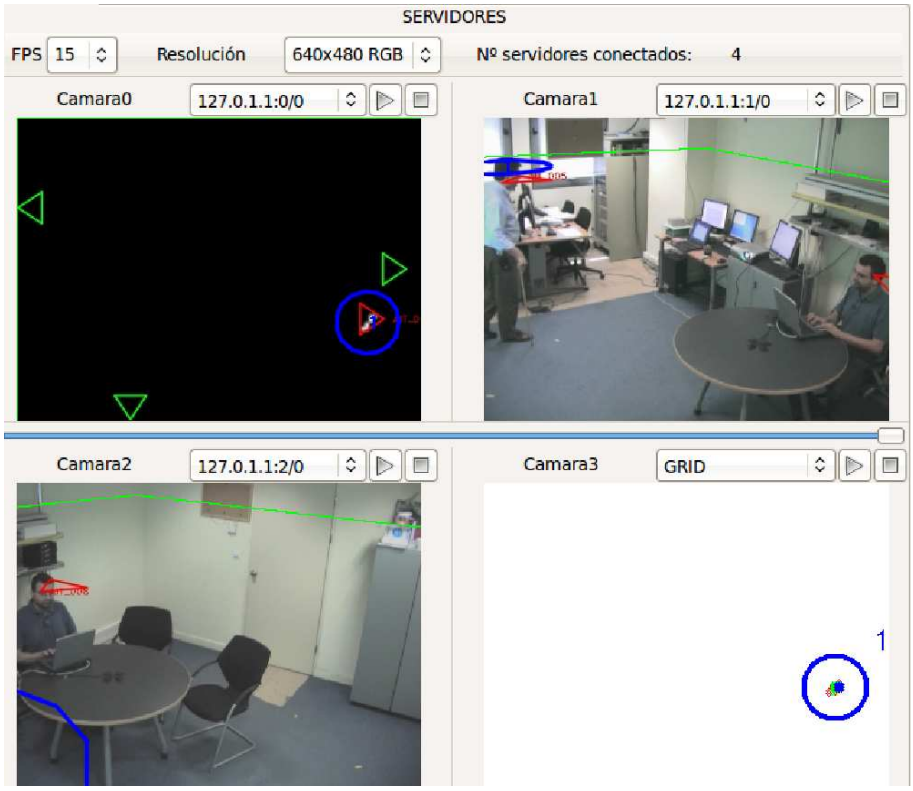


Figura 6.9: Escena con etiquetado acústico

En la Figura 6.10 existe una persona que es capturada al mismo tiempo por las tres cámaras, aunque sólo se muestran las imágenes de dos de ellas. Por lo tanto aparece grid procedente del sistema de visión ya que el grid de vídeo se forma realizando una AND del grid obtenido para cada una de las cámaras. De esta manera, si en una de dichas cámaras no aparece el objeto, su grid es nulo y el grid total será del mismo modo cero.

En la Figura 6.11 aparecen dos personas, siendo sólo una de ellas capturada por las tres cámaras a la vez, y por tanto, componiendo un grid para ella.

En la Figura 6.12 se encuentran las mismas dos personas que en la Figura 6.11. En este caso el Filtro de Partículas detecta a ambas, y esto es debido a que aunque una de ellas no sea capturada por todas las cámaras, se encuentra hablando en ese momento y es correctamente compuesto el grid procedente de información acústica. Por lo tanto, una de ellas tiene un grid procedente de información visual, y la otra de información acústica.

Situaciones observadas que suponen una mejoría en el sistema En este apartado se muestran aquellas situaciones en las que la inclusión del sistema de experimentación basado en audio ha supuesto una clara mejoría en los resultados obtenidos.

Como se puede observar en todas las escenas, aparece una persona que no es capturada al mismo tiempo por todas las cámaras, por lo que no se compone para ella grid procedente de visión. Sin embargo, bien sea por el ruido que se produce cuando entran en la sala (Figuras 6.13, 6.14, 6.16) o porque se encuentran hablando (Figura 6.15) se forma un grid a partir de información acústica que permite que el Filtro de Partículas las detecte. De esta manera, deja de ser condición fundamental para la detección de personas que sean capturadas por todas las cámaras, siendo posible dicha detección gracias a la inclusión del sistema de experimentación

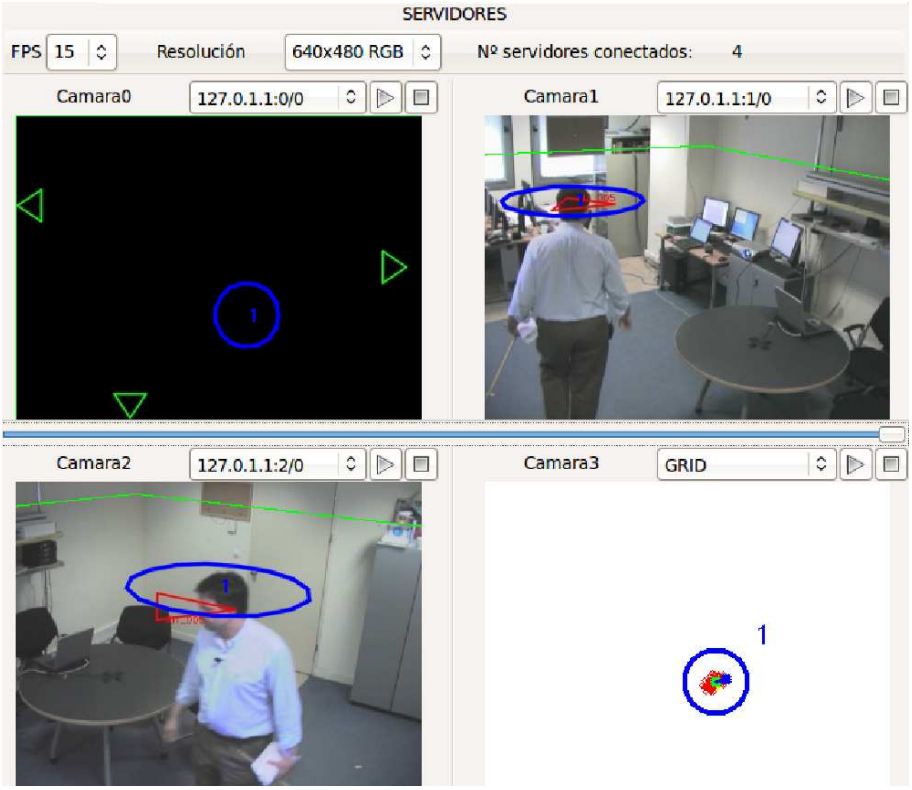


Figura 6.10: Escena con etiquetado procedente de visión

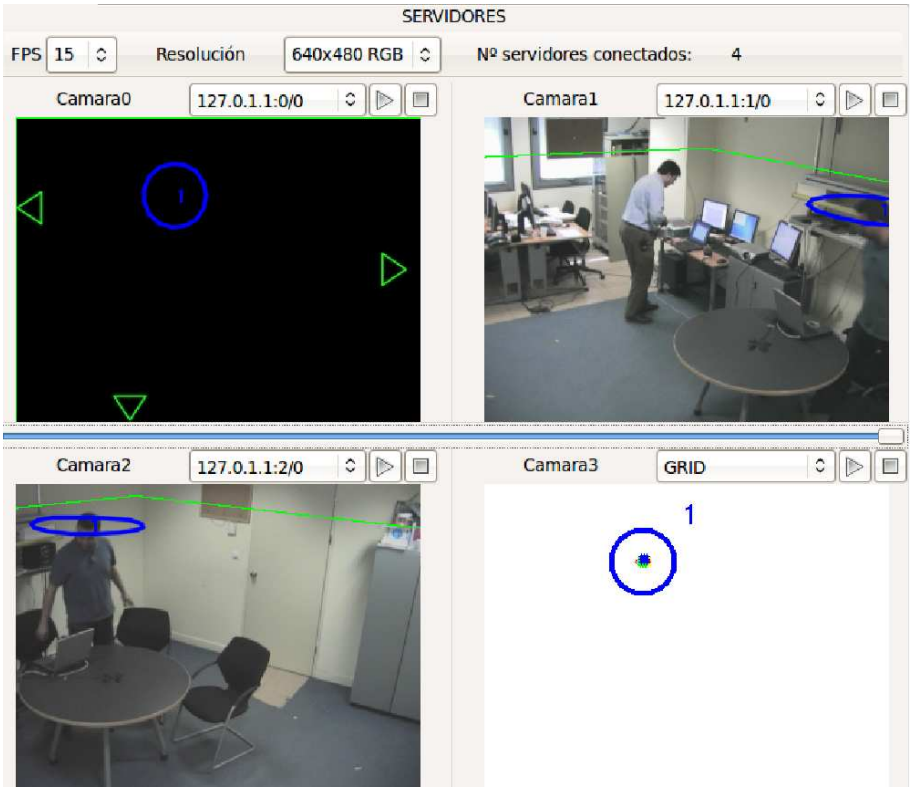


Figura 6.11: Dos personas en escena

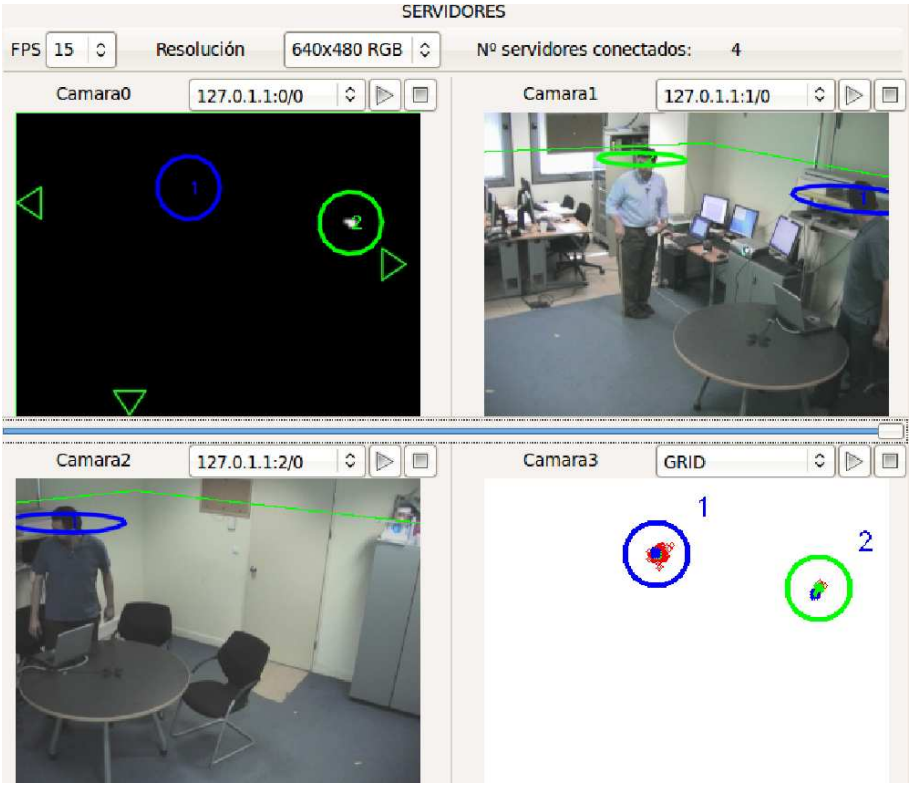


Figura 6.12: Dos personas en escena, ambas con grid validado

basado en audio.

Problemas encontrados en la detección de personas Se muestran a continuación situaciones en las que se encuentran problemas en la detección o en las que se producen diversos errores.

- En la Figura 6.17 aparece una persona en escena, siendo no detectada ya que sólo es capturada por una cámara de las tres que intervienen en el experimento, y además no se produce grid de audio que permita su localización, bien porque no existe ningún tipo de ruido o por un fallo en el sistema de experimentación de audio. En la Figura 6.18 se encuentra la misma situación pero en este caso se encuentran cuatro personas en la sala sin detectarse ninguna de ellas.
- En la Figura 6.19 aparecen cuatro personas en la sala, todas ellas con su posición etiquetada procedente del sistema de visión, y una de ellas del sistema de audio. En el sistema de visión, sólo la persona que está de pie (marcada con el número dos) es detectada ya que es la única que se encuentra por encima de la altura de 1700 mm prefijada para el experimento. Además, esta misma persona también es detectada gracias al grid de audio generado, siendo correcto ya que como se puede comprobar también aparece etiquetada su posición por encontrase hablando en ese momento. Sin embargo, si se observa la imagen de arriba a la izquierda, aparecen dos posiciones marcadas por el grid de audio (dos manchas blancas), siendo una de ellas erróneas (la número 1 en color azul), y llevando esto a la detección de un segundo elemento que no se encuentra en la escena.
- Anteriormente se ha mostrado que el sistema de experimentación acústico permite localizar personas que no son detectadas por todas las cámaras, suponiendo esto una enorme ventaja.

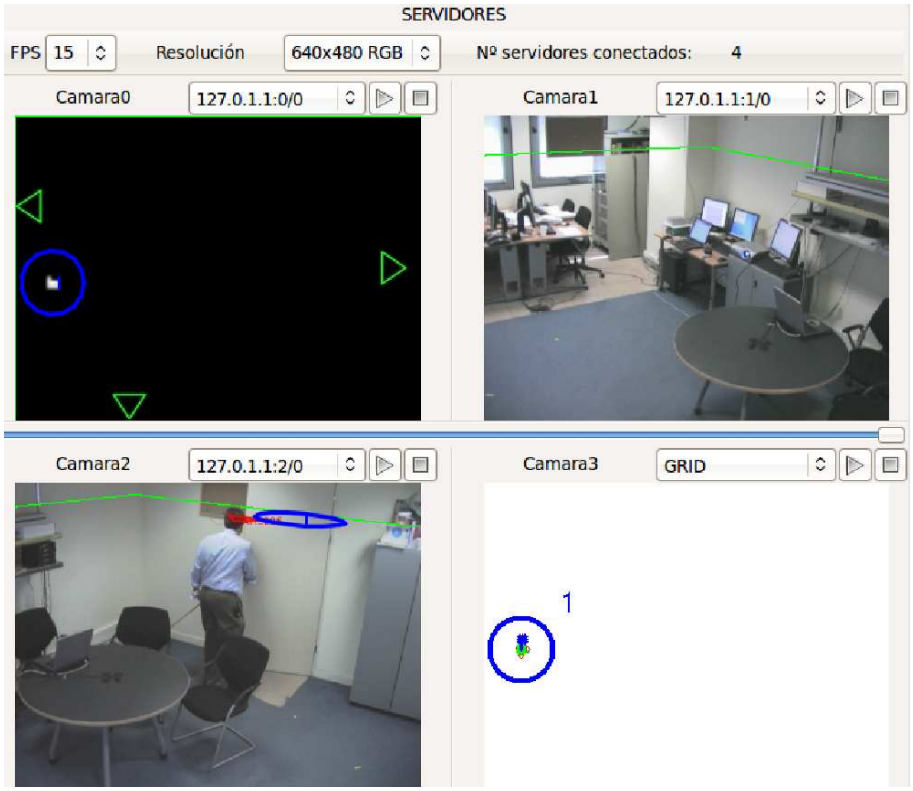


Figura 6.13: Ruido validado que permite detectar a una persona

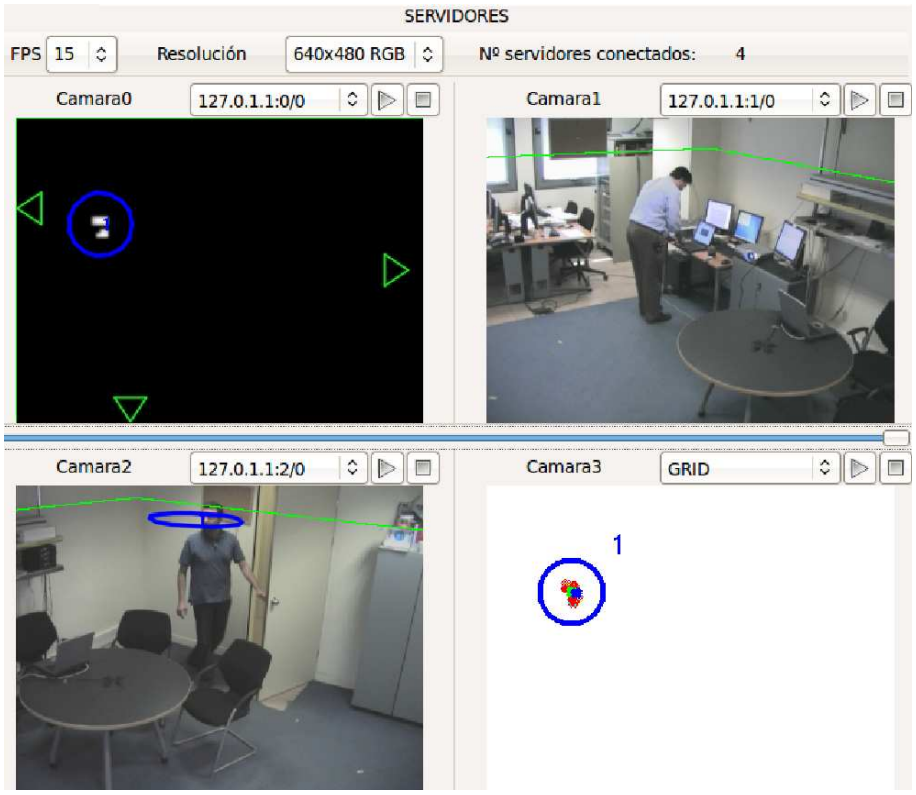


Figura 6.14: Ruido validado que permite detectar a una persona

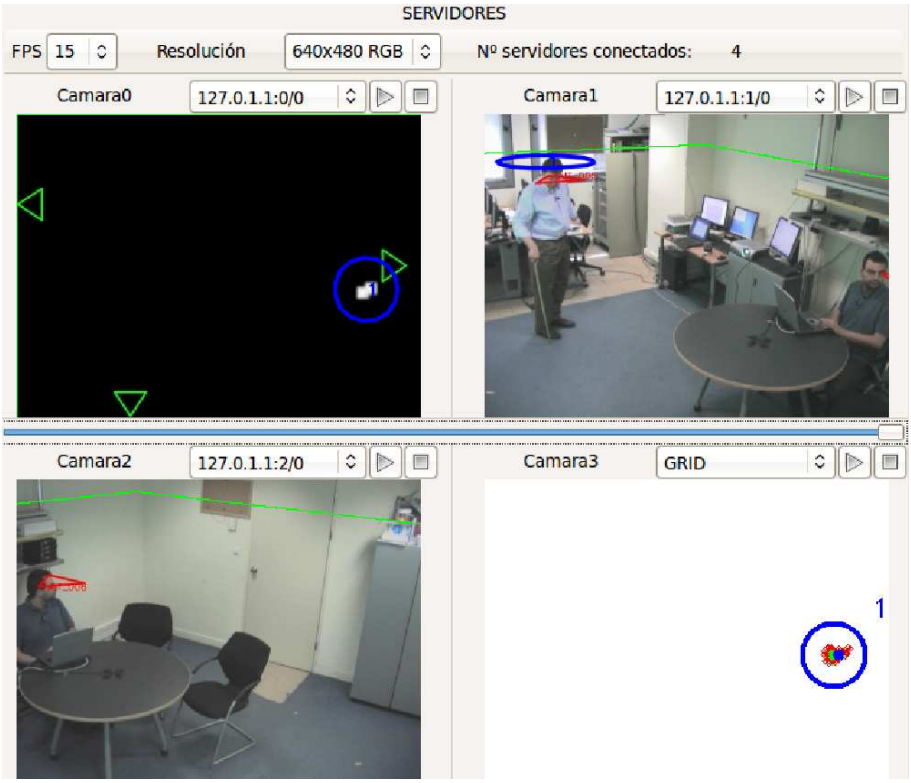


Figura 6.15: Ruido validado que permite detectar a una persona

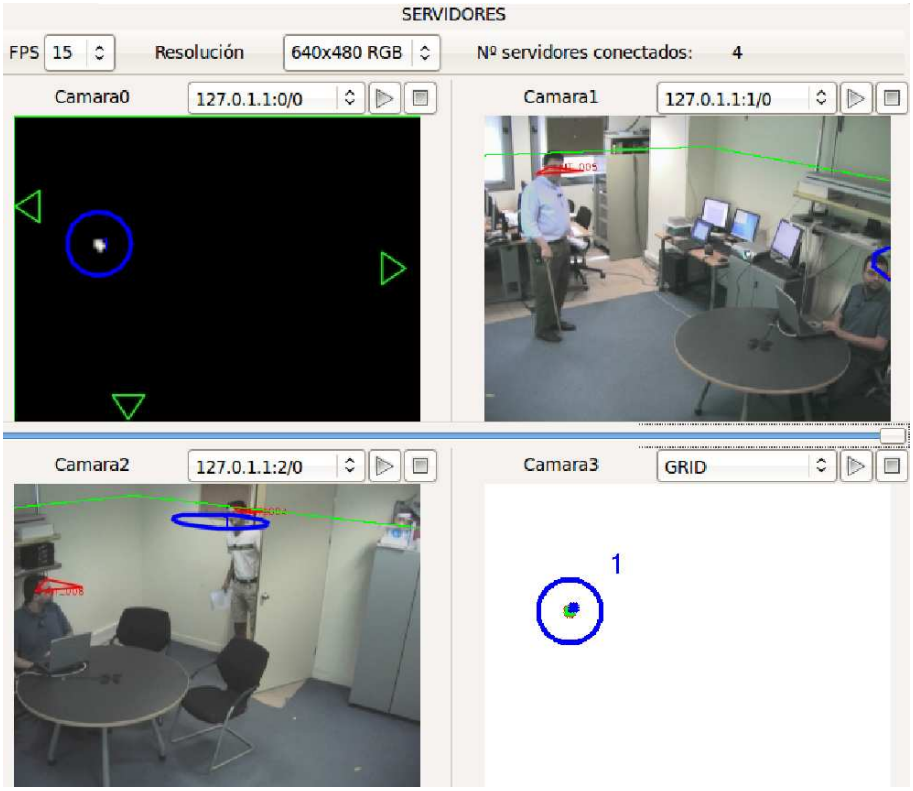


Figura 6.16: Ruido validado que permite detectar a una persona

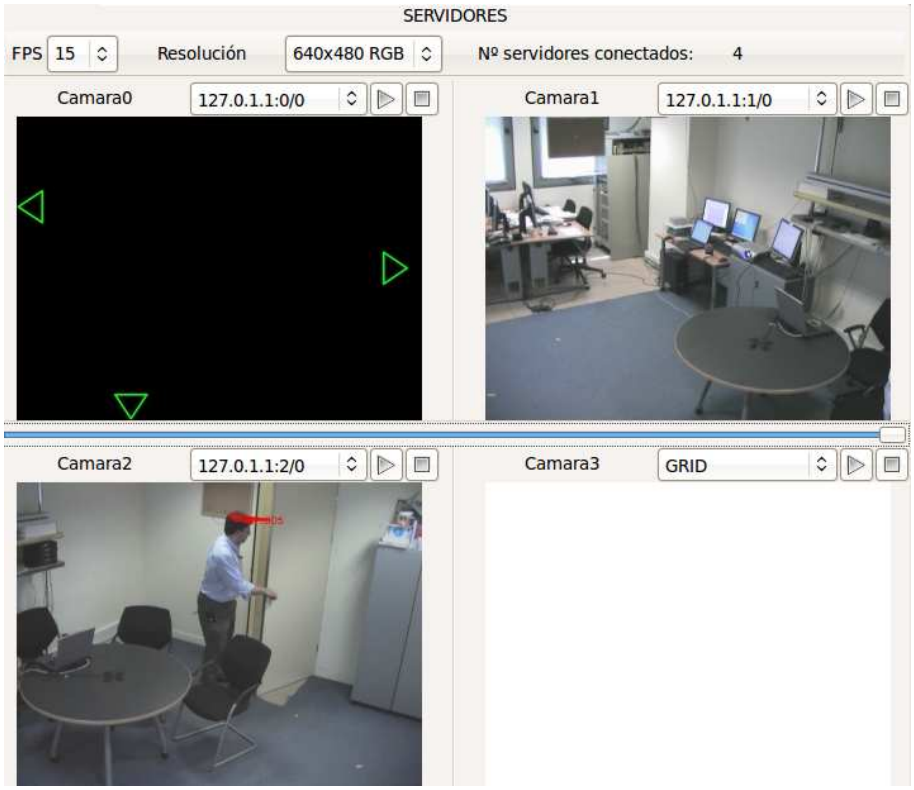


Figura 6.17: Persona sin detectar en la escena

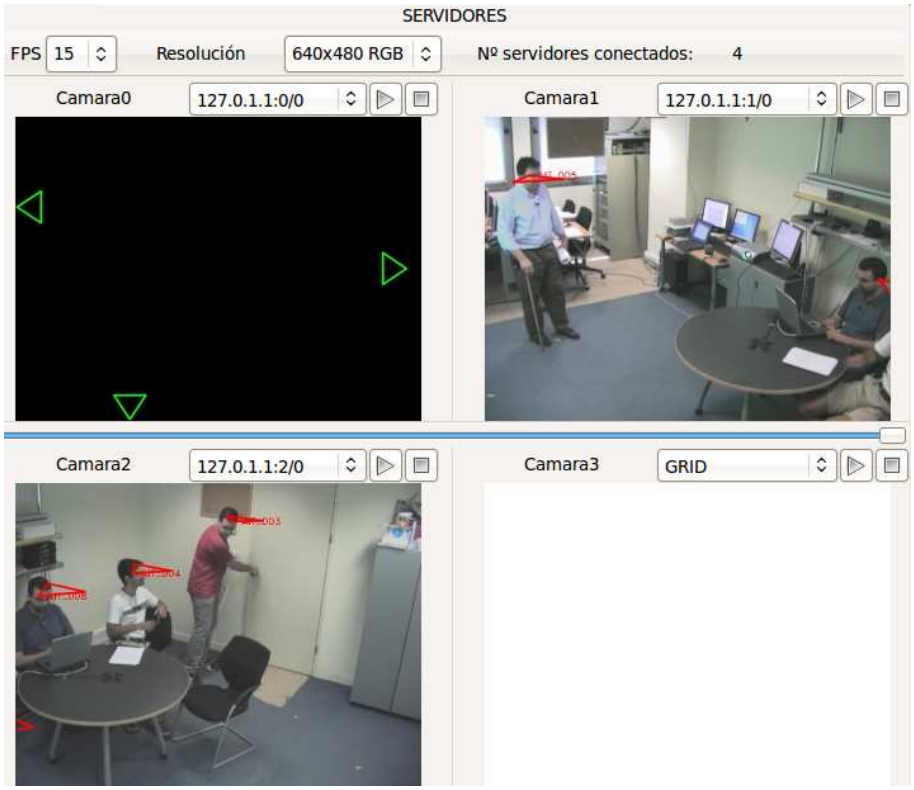


Figura 6.18: Personas sin detectar en la escena

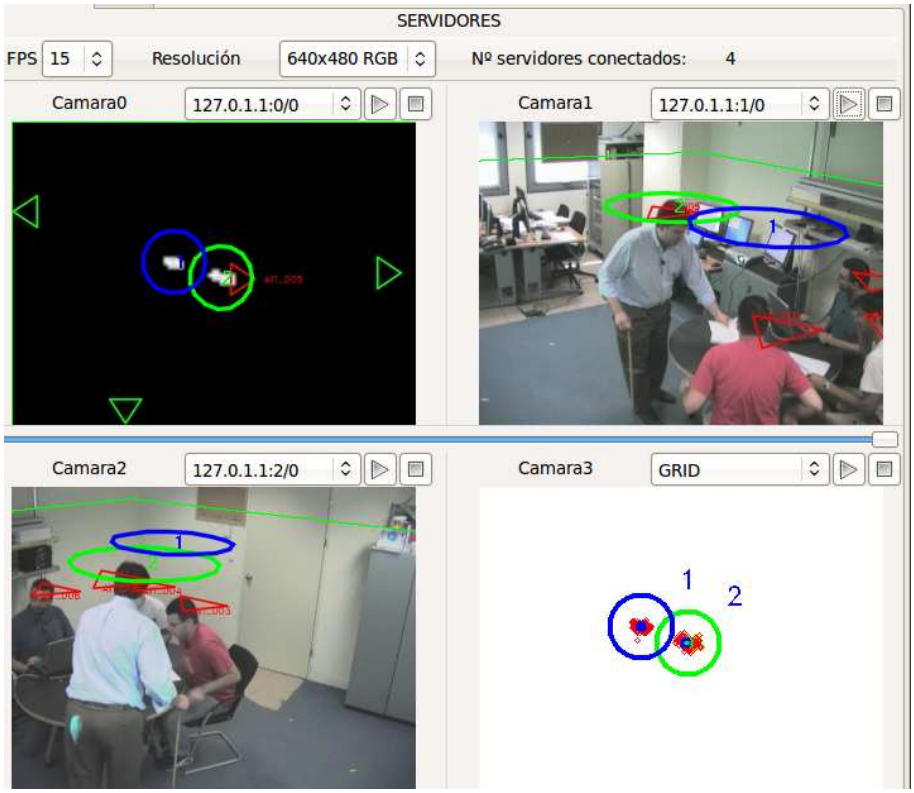


Figura 6.19: Error del sistema de experimentación basado en audio

Sin embargo, en ocasiones aparece grid de audio cuando únicamente hay un ruido (*toc toc* de la puerta), sin que ello suponga la existencia de una persona. En la Figura 6.20 se muestra en ejemplo de esta situación.

- Otro tipo de error consiste en la posición errónea del grid de audio, desviándose su posición de la real. En la Figura 6.21, 6.22, 6.23 se pueden ver estas situaciones y la distancia de sus correspondientes errores en centímetros.

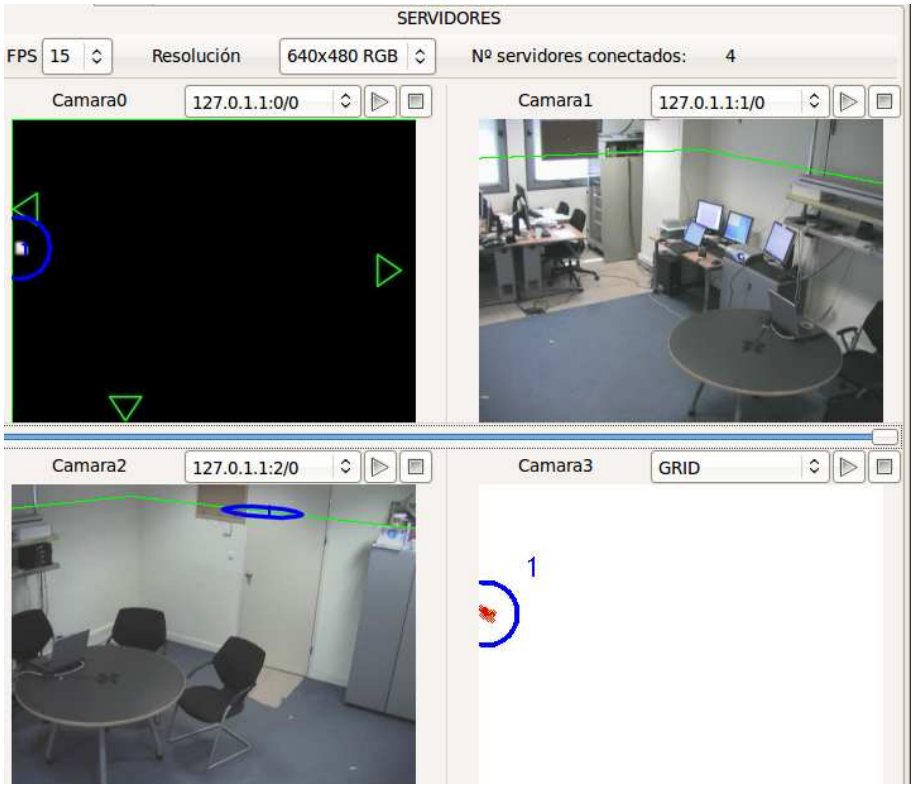


Figura 6.20: Detección de ruido

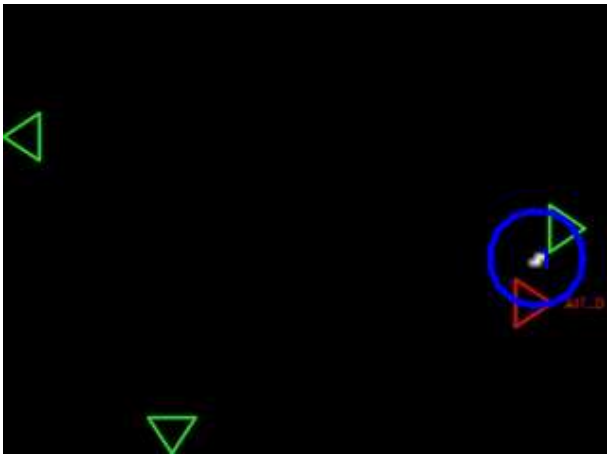


Figura 6.21: Error de 52 cm con respecto a la posición real

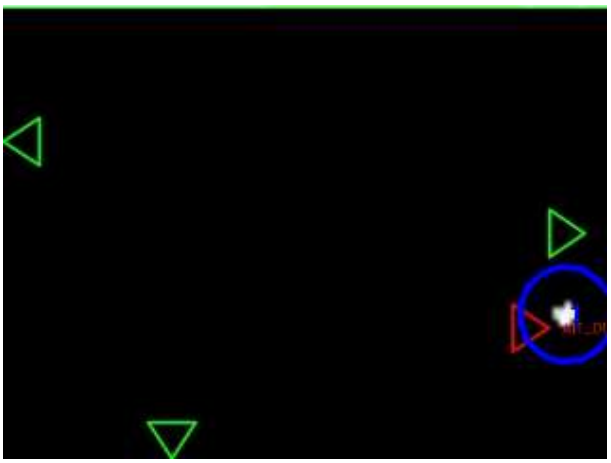


Figura 6.22: Error de 51 cm con respecto a la posición real

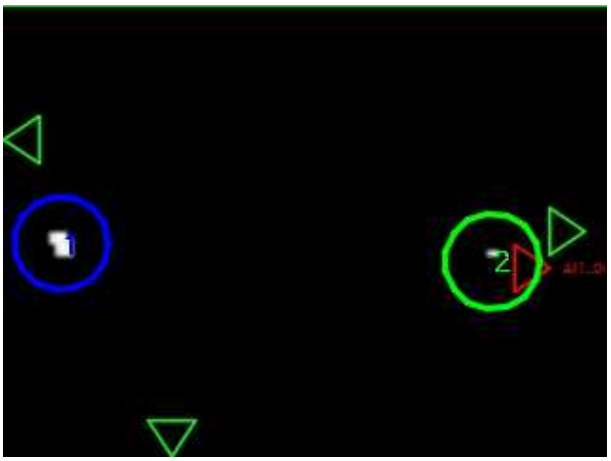


Figura 6.23: Error de 37 cm con respecto a la posición real

- Además, existen muchos instantes de tiempo donde no existen etiquetas, aunque se comprueba que sí aparecen personas presentes en la sala o existe alguien hablando, ver Figura 6.11. Esto puede ser debido a errores u omisiones en el etiquetado.

- Se ha explicado ya que el experimento se puede realizar prefijando una altura determinada. En este caso se ha establecido una altura de 1700 mm al considerar que es la estatura media de una persona que se encuentra de pie. Pero por ejemplo en el experimento mostrado en esta sección, la mayor parte del tiempo las personas se encuentran sentadas,no siendo entonces detectadas por el sistema de visión. En la Figuras 6.24 y 6.25 se muestra una misma escena con una altura de 1700 y de 1000 mm, comprobando que en el segundo caso la persona que se encuentra sentada sí es localizada.

6.4. Conclusiones

Se han observado los resultados obtenidos con el sistema de localización acústica, comprobando la tasa de error en cada uno de ellos. Se han mostrado las tasas de error para distintas bases de datos y distintas convocatorias CLEAR. Además dichas tasas se han representado en tablas

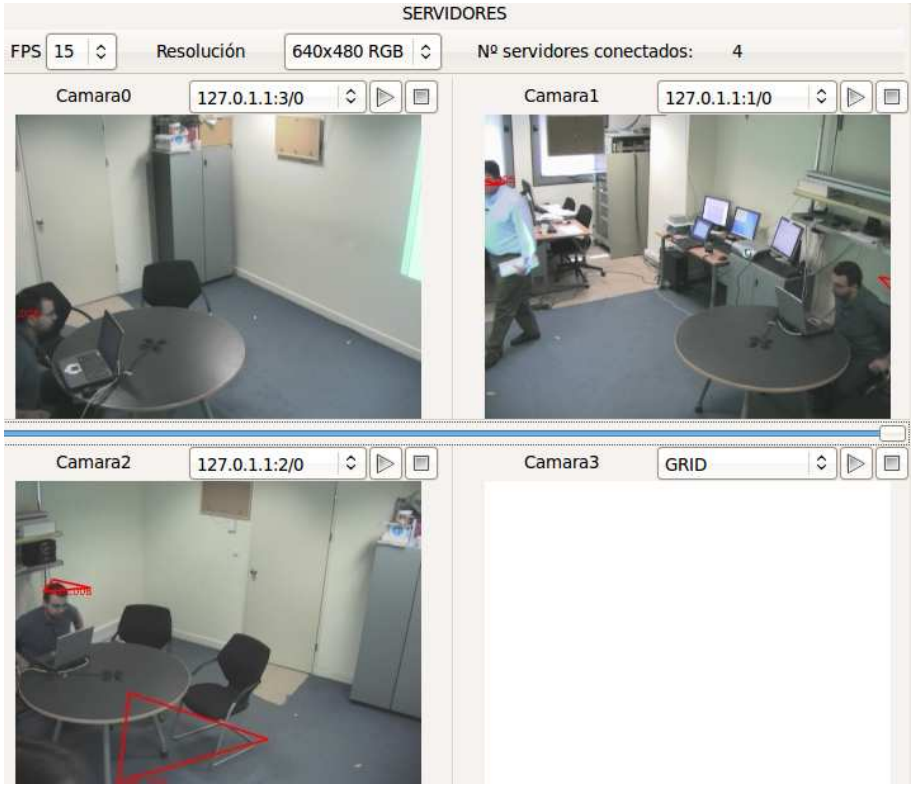


Figura 6.24: Escena a altura 1700 mm

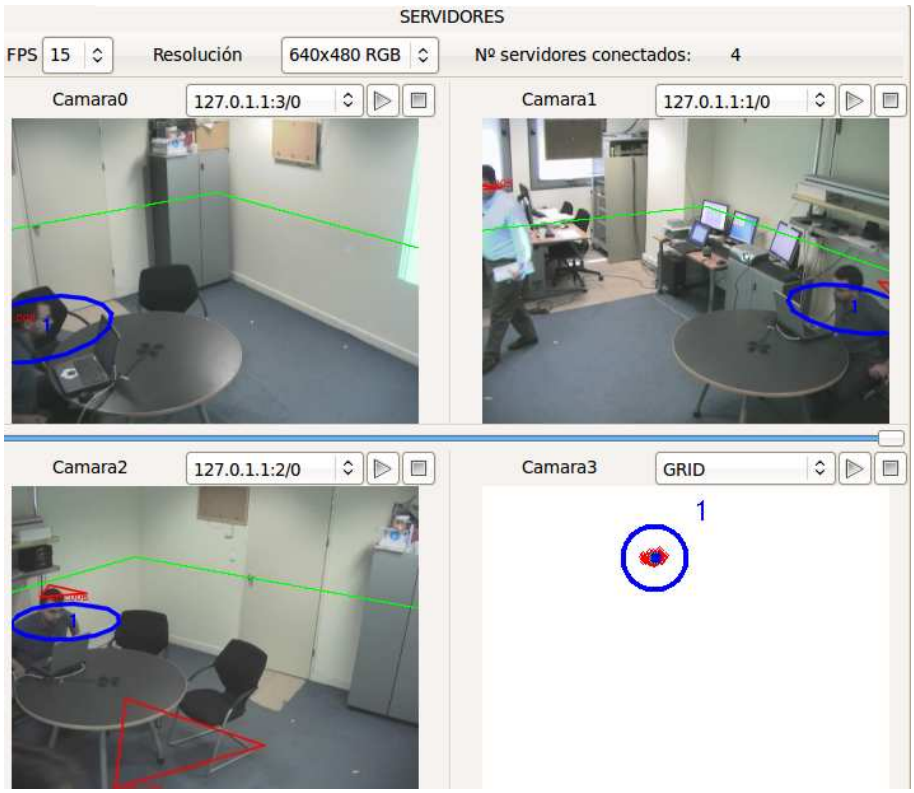


Figura 6.25: Escena a altura 1000 mm

comparativas entre distintas bases de datos o distintos experimentos, de manera que resulta sencillo comparar el error cometido en cada experimento.

Por otra parte, mediante multitud de imágenes se han expuesto distintas situaciones observadas en un experimento en concreto, situaciones que permiten conocer las bondades e inconvenientes del algoritmo de estimación basado en fusión audiovisual desarrollado.

Capítulo 7

Conclusiones y trabajos futuros

7.1. Introducción

Se exponen a continuación las conclusiones más relevantes tras la realización del trabajo completo relacionadas con los distintos sistemas de experimentación.

Además, se muestran una serie de líneas futuras que han surgido en la consecución de las fases del proyecto, ya que se han ido encontrando diferentes problemas, impedimentos, estrategias por validar, etc.

7.2. Conclusiones

Tras la realización de este trabajo se obtienen las siguientes conclusiones:

- Se ha mejorado un sistema de localización acústica, incorporando nuevas funcionalidades como el procesamiento de ficheros multicanal y de múltiples agrupaciones de micrófonos.
- Se ha conseguido un sistema de experimentación de procesamiento de audio versátil que permite realizar con facilidad experimentos sobre diferentes bases de datos.
- Se ha mejorado el sistema de experimentación basado en visión añadiendo prestaciones, implementando nuevas funcionalidades y mejorando las ya existentes. De esta manera se consigue una aplicación más versátil donde es posible implementar nuevos experimentos de manera sencilla.
- Se ha logrado implementar un sistema de estimación de la posición de objetos y personas basado en la fusión audiovisual. Esta fusión aporta ventajas frente al comportamiento sólo visual, ya que es posible localizar a las personas u objetos gracias a información acústica en instantes de tiempo en los que no se dispone de un grid procedente de visión.
- Resulta difícil construir un grid de ocupación acústica a partir de las imágenes obtenidas de los archivos de potencia generados por SRP, siendo dicho grid el que muestra la estimación de la localización de los hablantes o ruidos presentes en la sala.
- En el sistema de experimentación basado en audio e involucrado en la fusión audiovisual es necesario un método para discriminar el ruido de fondo de la voz humana y de otros ruidos puntuales. En este proyecto se propone una primera aproximación para caracterizar este ruido de fondo, de tal manera que se distingue del resto de señales existentes. El problema se encuentra a la hora de diferenciar la señal de habla humana de ruidos esporádicos con alta potencia que aparecen en el experimento, siendo esta tarea especialmente complicada.

7.3. Trabajos Futuros

Se proponen las siguientes líneas futuras:

- Como se ha comentado, cada sala tiene unas características diferentes que condicionan los resultados obtenidos mediante el algoritmo SRP. Por lo tanto, resulta interesante llevar a cabo un ajuste de los parámetros de entrada de manera individual para cada experimento, consiguiendo así los mejores resultados posibles en cada caso.

- Resulta necesario validar de algún modo la estrategia empleada en este trabajo para obtener el grid procedente de información acústica. Este método supone una primera aproximación, siendo conveniente su testeo para la posterior aplicación.
- Del mismo modo hay que validar la estrategia utilizada para validar grid de información acústica.
- Como se ha expuesto en las Conclusiones de este capítulo, el sistema de experimentación basado en fusión audiovisual se ha desarrollado intentando lograr en todo momento la versatilidad del mismo. Por lo tanto, sería más aconsejable que la totalidad de datos necesarios para lanzar un experimento pudieran ser proporcionados por línea de comandos o bien en un fichero de configuración inicial.
- Otro aspecto importante a mejorar en el sistema basado en fusión audiovisual es el desarrollo de un proceso que permita discriminar la potencia de los ruidos de aquella perteneciente a la voz humana. De esta manera, se evitarían los errores producidos en el algoritmo actual, donde el ruido de una puerta se trata de la misma manera que una señal de habla.
- Un trabajo pendiente es probar el algoritmo de fusión con diferentes experimentos de las bases de datos CHIL para comprobar diferencias en el comportamiento y realizar mejoras en el mismo. En este trabajo se han realizado pruebas únicamente con un experimento de una sala determinada.
- Se puede comprobar la bondad del sistema de fusión diseñado empleando los ficheros generados en formato CHIL con las posiciones de las clases validadas por el Filtro de Partículas para someterlos a una evaluación CHIL. De esta forma se podrán comparar los resultados obtenidos con los logrados por otros institutos o centros tecnológicos.
- Como ya se ha explicado, se dispone de un grid procedente de información visual y otro diferente de información acústica. Actualmente se realiza una OR lógica de ambos grids, no descartando la información de ninguno de ellos, y siendo tratados de la misma manera por el Filtro de Partículas. Una opción a probar sería asignar un peso diferente a las partículas pertenecientes al grid de audio o vídeo, pudiendo dar un mayor o menor peso a los dos tipos de grid.
- En el Espacio Inteligente del Departamento de Electrónica de la Universidad de Alcalá se encuentran instalados arrays de micrófonos y cámaras, por lo que se plantea como trabajo futuro la implementación en tiempo real del algoritmo de fusión.

7.4. Conclusiones

Se conocen así las principales conclusiones obtenidas tras la realización de este trabajo, aspectos clave del proyecto que se destacan tras el análisis completo de los desarrollos y resultados conseguidos.

Las líneas futuras dan idea de aquellas tareas de interés que se consideran importantes e interesantes realizar a partir de este trabajo.

Capítulo 8

Manual de Usuario

8.1. Introducción

Como se ha mostrado anteriormente, existen dos tipos de experimentos bien diferenciados. Por un lado el sistema de localización de hablantes basado en audio, y por el otro el sistema de experimentación basado en vídeo y procesamiento audiovisual.

Se muestran en cada caso datos relevantes para lanzar los experimentos, como son los parámetros de configuración, evaluación de los resultados... Finalmente, en el caso del sistema de audio se expone un ejemplo detallado de un experimento.

8.2. Manual de usuario del sistema de experimentación basado en audio

8.2.1. Introducción

Se parte de una situación en la que lanzar un experimento nuevo resulta una tarea tediosa y compleja ya que hay que tener en cuenta multitud de aspectos:

- En función del tipo de sala sobre la que se va a realizar el experimento ciertos parámetros han de ser modificados, personalizándolos de forma manual en cada ocasión (tamaño de la sala, áreas sin localización de hablantes...).
- El número de micrófonos que van a ser empleados así como sus posiciones dentro de la sala también son configurables para cada experimento.
- Es difícil llevar un control de los parámetros que intervienen en todas las etapas del proceso ya que se ecuentran repartidos en multitud de ficheros.
- El sistema de experimentación no cuenta con tareas que se realizan de forma automática, sino que el propio usuario debe generar distintos ficheros, realizar la llamada de otros, etc.

Por lo tanto lo que se desea es facilitar y automatizar todos los procesos que intervienen a lo largo del proceso de experimentación completo, siendo así posible realizar de manera sencilla experimentos de nuevas bases de datos.

El algoritmo `srp` está controlado por un conjunto de parámetros relevantes que condicionan los resultados obtenidos. Además se han desarrollado varios scripts para realizar diferentes tareas de manera automática, ya sea la preparación del experimento, la llamada al sistema de localización `srp`, la posterior evaluación de los datos, etc.

Se ha dedicado mucho esfuerzo a generalizar el sistema de experimentación; de este modo se consigue lanzar diferentes experimentos únicamente modificando ciertos parámetros de configuración.

8.2.2. Descripción del sistema de experimentación

Como se observa en la Figura 8.1 existen dos tipos de datos a tener en cuenta en el sistema de localización de hablantes:

- Datos de entrada: Se tiene un control total de estos datos de entradas para así poder describir cualquier escenario: tipo de habitación, geometría de los micrófonos, base de datos... Toda esta información se puede especificar en los siguientes ficheros:

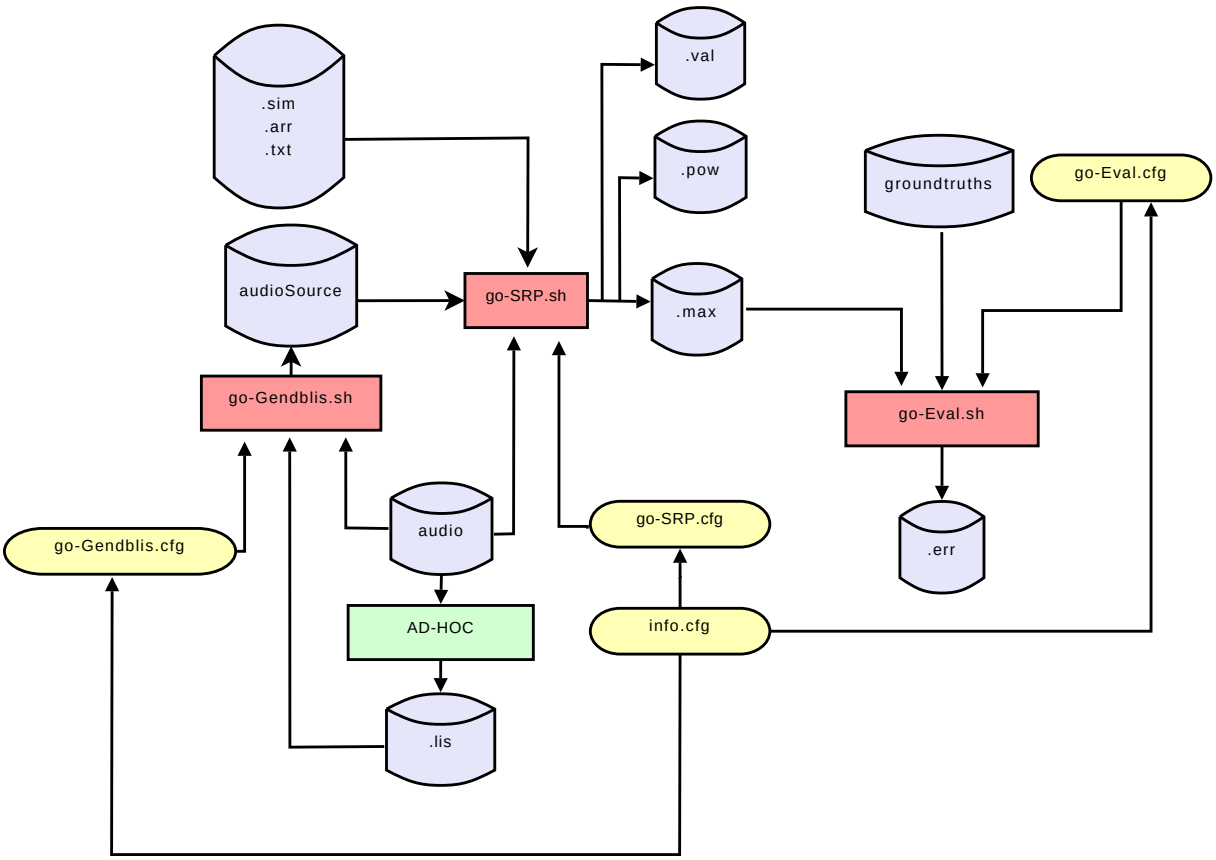


Figura 8.1: Diagrama del sistema de experimentación

- `roomName.sim`: Contiene un listado de todos los ficheros de configuración relacionados con la geometría del entorno, así como los directorios en los cuales se pueden encontrar. Hay que señalar que en este aspecto existe una limitación, y es que las salas se consideran paralelepípedos desde el punto de vista de espacio de búsqueda soportado.
  - `micArray.arr`, en este fichero se especifican los micrófonos que van a ser empleados en el experimento. Además se proporcionan las coordenadas (x,y,z) de cada uno de estos micrófonos en milímetros.
  - `subMicArray.subarr`, en ocasiones resulta interesante obtener resultados realizando agrupaciones de micrófonos, para así estudiar diferentes comportamientos del algoritmo `srp`. En el caso de que se empleen subarrays de micrófonos, un fichero `.subarr` debe proporcionarse en el archivo de simulación `.sim`. Es importante destacar que en este caso no debe aparecer ningún `micArray.arr` en dicho archivo. En el fichero en el que se detallan estos grupos de micrófonos debe especificarse el nombre de un `micArray.arr` que debe contener el listado de la totalidad de micrófonos involucrados en el experimento, así como sus posiciones (x,y,z). Más adelante, en la sección 8.2.4 se muestra como ejemplo un experimento en el que se han contemplado ambas posibilidades, tanto el uso de grupos de micrófonos como el empleo de la totalidad de ellos sin ningún agrupamiento.
  - `searchSpace.txt`, en este fichero se detallan los límites espaciales del entorno en el que se va a localizar los hablantes. Dichos límites se proporcionan en coordenadas (x,y,z). Además de esta información se proporciona también un espaciado en las tres dimensiones que el algoritmo emplea para separar consecutivos puntos de búsqueda. La unidad escogida es, al igual que en los anteriores casos, el

milímetro.

- `deadAreas.txt`, contiene la especificación de áreas muertas como aquellas en las que no tiene interés que el programa realice la búsqueda de locutores. Estas áreas se definen mediante las coordenadas (x,y) de sus límites. Ejemplos de áreas sin interés son armarios, mesas grandes, etc.

- `audiosource-database`, especifica los ficheros que contienen el audio de los canales que se van a procesar para todas las secuencias de experimentación. La primera línea de este archivo contiene dos valores a emplear por el programa. En primer lugar el número de archivos a tener en cuenta para cada seminario (`NUMFILESPERUTT`), y en segundo lugar el número de caracteres necesarios para construir los diferentes nombres de los ficheros de salida de cada seminario (`SIZECOMMONFILENAME`).

- Datos de salida: Contienen las posiciones estimadas en la localización de hablantes calculadas por el algoritmo `srp`. Empleando estos datos es posible evaluar la exactitud en las estimaciones, comparándolos con las posiciones reales o *groundtruth*.

  - `baseName.max`, este fichero está formado por diferentes columnas. En la primera de ellas se especifica el índice de tiempo en segundos para el que se va a proporcionar la estimación. Las posteriores columnas se encuentran agrupadas de 3 en 3; conteniendo cada uno de estos grupos las posiciones (x,y,z) en milímetros donde el sistema estima que se va a encontrar el locutor.
  - `baseName.val`, este fichero contiene la totalidad de los índices de tiempos expresados en segundos para los que se ha realizado una estimación. Para cada uno de estos tiempos se detalla la potencia media del frame expresada en dB. Posteriomente se listan tantas coordenadas (x,y,z) como número de máximos se hayan especificado en la correspondiente opción de la línea de comandos.
  - `baseName.pow`, este fichero se obtiene como salida si se asigna un valor '1' a la variable `PRINT_POW_LOC` que se puede encontrar en el archivo `srp.c`. Éste es un fichero donde se muestra, para cada instante de tiempo y cada punto de búsqueda del espacio, la potencia de audio calculada.
  - `baseName.tree`, este fichero sólo se genera en caso de que la versión coarse-to-fine del srp se ejecute. Los datos contenidos son los mismos que el `baseName.val` con la diferencia de que no sólo se proporciona la posición final estimada para cada índice de tiempo, sino todas las posiciones previas estimadas de acuerdo a la metodología coarse-to-fine (ver la sección Estrategias adicionales de la página 49).

Con el fin de simplificar la generación de nuevos experimentos se han desarrollado una serie de bash scripts y archivos de configuración, cuyos nombres se observan en la Figura 8.1. De este modo se obtiene un sistema de experimentación genérico, donde únicamente es necesario modificar una serie de parámetros y campos para lanzar un nuevo experimento.

La estructura general del procedimiento de experimentación ha sido diseñada atentiendo a diferentes etapas como son:

- Tareas y configuración de generación de la estructura principal del experimento.
- Tareas y configuración dependientes de la base de datos.
- Tareas y configuración dependientes del experimento.

- Tareas y configuración dependientes de la evaluación.

Todos los archivos están almancenados en el directorio `autoGenExp`, bajo la estructura de directorios general que contiene el software relacionada con tareas de audio empleando arrays de micrófonos. Algunos de los ficheros se han almacenado en el directorio `skeleton` localizado dentro de `autoGenExp`. Además, un fichero `README.autoGenExp` con instrucciones básicas está disponible. Los archivos de configuración y los scripts se detallan a continuación:

- `info.cfg`: Este fichero contiene información de carácter general que es necesaria en todos los scripts, por ejemplo la fecha, nombre de usuario,etc.
- `genExp.cfg`: Se proporcionan las rutas de directorios implicados en el sistema de experimentación: home, experimentos, srp.
- `genExp.sh`: Mediante la ejecución de este fichero se crea el directorio del nuevo experimento dentro de `experiments`, y se almacenan dentro de él todos los archivos de configuración y scripts que serán necesarios posteriomente para la correcta ejecución y evaluación.
- `genExp.err`: Diferentes mensajes de error son mostrados si algún error se produce en la generación del directorio del experimento.
- `db_id.cfg`: Contiene el nombre asignado por el usuario al experimento que va a realizar, y empleado posteriormente en diferentes scripts.
- `go-GENDBLIS.cfg`: Se especifican definiciones relacionadas con la base de datos.
- `go-GENDBLIS.sh`: Este script genera el archivo de entrada `audiosource` basado en un fichero `.list`. El `audiosource` contiene un listado con todos los archivos de audio de una base de datos determinada involucrados en el experimento a realizar.
- `go-SRP.cfg`: Se detallan los parámetros de ejecución de `srp`.
- `go-SRP.sh`: Este script realiza la llamada a `srp` con los parámetros apropiados y enlazándolo con los correspondientes ficheros de entrada (archivo de simulación `.sim` y `audiosource`).
- `go-EVAL.cfg`: Parámetros necesarios para evaluar correctamente los resultados ofrecidos por `srp`.
- `go-EVAL.sh`: Este script llama a la aplicación de evaluación correspondiente a los estándares de CHIL, con el objetivo de evaluar los resultados obtenidos mediante el algoritmo.

8.2.3. HOWTO del sistema de experimentación

A continuación se muestra de forma detallada cada uno de los pasos para lanzar un nuevo experimento:

8.2.3.1. Tareas y configuración de generación de la estructura principal del experimento

En primer lugar es necesario configurar con unos sencillos pasos el comportamiento general del experimento a diseñar:

1. Añadir en el fichero `.bashrc` del `home` del usuario la constante `FAR_FIELD_ROOT` con la ruta en la que se encuentra toda la distribución a emplear. Existe también la posibilidad de suministrar el valor de esta constante en el archivo de simulación `.sim` o bien por línea de comandos en la llamada a `srp`. El programa en primer lugar comprueba en el `.sim`, buscando la variable `dirRootDistrib`; en caso de no encontrarla pasa a buscarla en la opción de la línea de comandos del `srp` llamada `dir-root-distrib`; como última opción el sistema acude a la variable de entorno:
`FAR_FIELD_ROOT="$HOME/repositorio/far-field"`

2. Crear un fichero de configuración con un identificador del experimento `EXP_ID` y un objetivo `OBJECTIVE`. El formato recomendado para el identificador de experimento es `<BASE DE DATOS>-<EXPERIMENTO>`. Este archivo de configuración debe ser guardado en el directorio `exp_cfg_files` localizado en el directorio `autoGenExp`.

3. Ejecutar `genExp.sh` seguido del nombre completo del `.cfg` creado en el paso anterior.

4. En este momento se habrá creado un directorio con el nombre identificativo establecido (`EXP_ID`) debajo de `far_field/experiments`. . Este nuevo directorio contiene los siguientes archivos:
`info.cfg`
`go-GENDBLIS.sh`
`go-GENDBLIS.cfg`
`go-SRP.sh`
`go-SRP.cfg`
`go-EVAL-sh`
`go-EVAL.cfg`
`genExp.cfg`
`README`
`README.autoGenExp`

8.2.3.2. Tareas y configuración dependientes de la base de datos

1. Es necesario crear el fichero `db_id.cfg`, donde se especifica la variable `DB_ID` que es el nombre asignado al experimento y que es empleado en diferentes scripts.

2. En el directorio del experimento hay que chequear las variables definidas en `go-GENDBLIS.cfg`. Es necesario establecer los valores correctos de `SIZECOMMONFILENAME, NUMFILESPERUTT, NUM_MICS y CHANNELS`, así como confirmar mediante la variable `DB_HOME_DIR` la ruta en la cual se encuentran los ficheros de audio de la base de datos tenida en cuenta.

3. El número de ficheros por cada seminario o segmento listado en el `audiosource` es proporcionado en `NUMFILESPERUTT`; de esta forma se permite un funcionamiento correcto con ficheros de audio monocanal o multicanal. En el ejemplo mostrado en la página 146 se pueden encontrar detalles relacionados con este parámetro. El número de micrófonos empleados en el experimento se proporciona en la variable `NUM_MICS`. Por último `SIZECOMMONFILENAME` establece el número de caracteres comunes en los ficheros de audio involucrados, es decir, tamaño del nombre a partir del cual se va a generar el nombre del fichero de salida. Por último, si se va a realizar un experimento donde intervienen ficheros de audio multicanal se debe especificar en `CHANNELS` los canales específicos que van a ser empleados, por ejemplo: `CHANNELS=( 1 2 3 4)`. Se puede comprobar el funcionamiento de esta

utilidad en el ejemplo de la página 146. Si por el contrario sólo se van a emplear ficheros de audio monocanal, se debe dejar vacía la variable de tal forma: `CHANNELS=()`.

4. En todos los experimentos se debe generar un fichero `.list` conteniendo un listado de todos los archivos de audio que se van a emplear. Este fichero debe estar localizado en el directorio del experimento que se va a evaluar, y su nombre debe ser el mismo que el del identificador del experimento, de otra manera, se obtendrá un error.

5. Al ejecutarse el script `go-GENDBLIS.sh` se creará un fichero `audiosource` con el siguiente formato: `<audiosource>-<DB_ID>`. Este `audiosource` contiene los paths relativos a cada uno de los ficheros de audio que tomarán parte del experimento. Éste es un script ad-hoc, por lo que si una nueva base de datos se quiere evaluar es necesario añadir líneas de código en `go-GENDBLIS.sh` para generar el `audiosource`, ya que los paths y nombres de archivos de audio son diferentes en cada base de datos.

8.2.3.3. Tareas y configuración dependientes del experimento

1. En el fichero de configuración `go-SRP.cfg` se deben confirmar parámetros necesarios por el algoritmo `srp.c`: `FRAME_SIZE_SECS, FFT_SIZE, SIM_FILENAME, FS`..., prestando especial atención en los parámetros establecidos como obligatorios y aquellos que son opcionales. El fichero de simulación `.sim` debe estar localizado en el mismo directorio del experimento. A continuación, un extracto de un archivo de simulación sin el uso de subarrays se muestra:

   ```
   [MicArrays]
   numMicArrays = 2
   micArray0 = leftCircularArray8.arr
   micArray1 = rightCircularArray8.arr
   ```

   Si se requiere el uso de subarrays, el archivo de simulación debe ser parecido al que se muestra a continuación, donde sólo aparece un único `.subarr`:

   ```
   [MicArrays]
   numMicArrays = 1
   micArray0 = subArrays.subarr
   ```

   En el `.subarr` deben especificarse el número de subarrays, el número de micrófonos de cada subarray y los índices de los micrófonos que intervienen en cada subarray. Estos índices están relacionados con la posición de cada micrófono en el fichero `.arr`. Además, cada `.subarr` apunta a un `.arr` que contiene todos los micrófonos empleados y sus posiciones correctas. A continuación se muestra una posible configuración de un fichero de subarrays:

   ```
   numSubArrays = 2
   [MicArray]
   dirMicArrays = /u/maria/respositorio/proyecto/far-field/environments/IdiapRoom
   micArray = allCircularArray.arr
   [SubArrayInfo]
   micsPerSubArray0 = 8
   micsPerSubArray1 = 8
   sub0 = 0 1 2 3 4 5 6 7
   sub1 = 8 9 10 11 12 13 14 15
   ```

   Se proceden a explicar de forma breve los parámetros necesarios por el algoritmo `srp.c`, aunque toda la información se puede encontrar tecleando `./srp -help` en el directorio donde se encuentre el programa srp.

- `FS`: La frecuencia de muestreo `FS` sólo se empleará en el caso de ficheros de audio puros. De otra forma el programa obtendrá la frencuencia leyendo el fichero de audio y la considerará como aquella a la que fue grabado.
- `FFT_SIZE`: El tamaño de la FFT debe ser siempre mayor o igual al tamaño del frame `FRAME_SIZE_SECS`. Además, y por razones meramente computacionales, debe ser potencia de 2.
- `DIR_INPUT_FILES`: Esta opción de la línea de comandos especifica el directorio en el cual el programa puede encontrar los ficheros requeridos de entrada, es decir, el `audiosource` y `.sim`.
- `INI_AUDIO_FILE, END_AUDIO_FILE`: Determina el límite de tiempo en segundos de la pieza de audio que va a ser analizada. Como es obvio, estos valores deben ser positivos y siempre `END_AUDIO_FILE` debe ser mayor que `INI_AUDIO_FILE`. Si el parámetro `END_AUDIO_FILE` se establece como un tiempo mayor al total de la duración del archivo de audio, automáticamente se limitará al tiempo total en segundos. Por último, si se desea analizar la duración total de los archivos de audio, ambos parámetros se especifican a 0.
- `FREQ_SRP_FLAG`: Cuando se activa este parámetro, el método FSRP (Frequency SPR) será empleado para realizar el experimento. Como consecuencia de ello, el parámetro `ROUND_FLAG` deja de tener efecto ya que únicamente influye en el caso de emplearse el método TSRP (Time SRP). Si `FREQ_SRP_FLAG` se encuentre desactivado, se asume que se utiliza el método TSRP.
- `ROUND_FLAG`: Puede tener los siguientes valores: 0 (no se aplica ningún redondeo), 1 (se aplica un redondeo al entero más próximo) o >1 (se aplica un redondeo acorde a una interpolación lineal).
- `LOW_FREQ`: Su valor puede ser empleado en dos casos diferentes: 1) Si el `FILTER_FLAG` está activado marca la frecuencia baja de corte, y 2) si el flag `COARSE2FINE` está activado marca la frecuencia de corte inicial en el primer paso del esquema *coarse to fine*.
- `MAX_DIST`: Sólo se aplica cuando el método *coarse to fine*está activo.
- `FIXED_THRESHOLD_FLAG`: Si está activo (dándole un valor de 1) se emplea un límite fijo de máscara de ruido, si no (valor de 0) se empleará un límite adaptativo. Sin embargo, este límite sólo tendrá consecuencia en el caso de que el Flag de Máscara de Ruido haya sido activado.
- `NOISE_SIZE_SECS`: Especifica, en segundos, la longitud del frame localizado al principio del archivo de audio sobre el que se llevará a cabo la estimación de ruido. Como es de suponer, este parámetro sólo influirá en el caso de que el Flag de Máscara de Ruido se encuentre activo.
- `NOISE_THRESHOLD`: Establece el número de dB que la señal de audio debe tener para sobrepasar el nivel de ruido y no ser descartada. Como los anteriores, este parámetro sólo tendrá influencia en el caso de que el Flag de Máscara de Ruido esté activo.
- `CHANNELS`: Este comando especifica los canales empleados en el caso de un experimento multicanal. El formato es `CHANNELS="6,7,8,..."` si los ficheros de audio son monocanal o `CHANNELS=´´all"` si se quieren utilizar todos los canales del fichero de audio.

2. Se ejecuta `go-SRP.sh`. Un fichero `.log` con información relacionada con la ejecución del programa y posibles errores será creado. Además, un fichero llamado `<EXP_ID>.run` se

crea la primera vez que se ejecute el script. La fecha en la cual `go-SRP.sh` fue lanzado y la lista de argumentos utilizados en cada experimento se almacenan en este fichero; por lo tanto, en posteriores ejecuciones el archivo se va actualizando. El formato del fichero `.log` con toda la información de ejecución es `<fecha&hora>-<EXP_ID>.log`.

8.2.3.4. Tareas y configuración dependientes de la evaluación

1. Para evaluar los resultados del algoritmo `SRP`, en primer lugar se deben confirmar los siguientes campos en el fichero `go-EVAL.cfg`: `EVALAPP y OVERALLAPP` .
2. Se ejecuta `go-EVAL.sh`. Un fichero `.err` se crea, conteniendo los resultados de la evaluación acorde a los estándares CHIL. También se recopila información sobre la ejecución de la evaluación en un fichero `.log`. El formato del archivo `.err` es `<fecha&hora>-<EXP_ID>-chilError.err`. El formato del fichero `.log` es `<fecha&hora>-<EXP_ID>-eval.log`. Además un `<EXP_ID>-eval.run` se crea la primera vez que se lleve a cabo la evaluación, y se actualiza cada vez que se ejecute `go-EVAL.sh`. La fecha en la que se lanzó la evaluación y la lista de argumentos de cada evaluación realizada se almacenan en este fichero.
3. Después de la evaluación completa de los resultados, es posible evaluar sólo algunas secuencias de interés. Para hacer esto se debe crear otro `.list` con las secuencias elegidas. Se ejecuta entonces `go-EVAL.sh` seguido del nombre del fichero `.list` (sin escribir `.list`). Al realizar este paso se crean sus propios ficheros `.err` y `log`.

8.2.4. Ejemplo

En esta sección se va a mostrar un ejemplo real con el fin de lanzar un experimento completo, en concreto empleando una base de datos HIFI.

8.2.4.1. Tareas y configuración dependientes del experimento

1. Generar el directorio del experimento y sus archivos correspondientes.
   - En primer lugar, se debe confirmar el directorio raíz en `genExp.cfg`, por ejemplo: `HOME_DIR=´´u/maria/repositorio/proyecto"`.
   - Se crea un archivo de configuración, en este caso se llama `sampleExp.cfg` y se guarda en el directorio `autoGenExp/exp_cfg_files`. El valor de `EXP_ID` y `OBJECTIVE` es:
     `EXP_ID="HIFI-MM1-samplerate"`
     `OBJECTIVE="Testear SRP con HIFI-MM1 comprobando la dependencia con samplerate"`
     Donde el formato de `EXP_ID` debe ser, como se explicó anteriormente, `<BASE DE DATOS>-<EXPERIMENTO>`.
   - Se ejecuta `genExp.sh` seguido por el archivo de configuración:
     `$ bash genExp.sh sampleExp.cfg`
     Ahora, en el directorio `experiments` aparece un nuevo directorio llamado `HIFI-MM1-samplerate`, exactamente como `EXP_ID`.

8.2.4.2. Tareas y configuración dependiente de la base de datos

1. Se genera el archivo `audiosource` que contiene los ficheros de audio de interés y se configura el conjunto de ficheros de audio que se emplearán en el experimento. A partir de ahora

se trabajará con los ficheros almacenados en el directorio del experimento, es decir, en el directorio `HIFI-MM1-samplerate`.

- Se genera el fichero `db_id.cfg` con el siguiente contenido:
  `DB_ID="HIFI-MM1-samplerate"`
- Se escribe en un fichero de texto llamado `HIFI-MM1-samplerate.list` los nombres de los ficheros de audio que se van a emplear en la localización de la siguiente manera:
  ```
  LFD-P1-O1-U11413
  LFD-P1-O1-U11421
  LFD-P1-O1-U15003
  ...
  ```
  Este fichero `.list` debe localizarse dentro del directorio del experimento, si no es así aparecerá un error.
- Se confirman los parámetros necesarios en `go-GENDBLIS.cfg`, en este ejemplo serían:
  ```
  SIZECOMMONFILENAME=15
  DB_ID="HIFI-MM1"
  DB_HOME_DIR="$HOME_DIR/corpora/speech-split/LFD"
  NUM_MICS=4
  NUMFILESPERUTT=3
  CHANNELS=( 3 4 )
  ```

- Se ejecuta `go-GENDBLIS.sh`:
  `$ bash go-GENDBLIS.sh`
  El archivo `audiosource`, llamado `audiosource-HIFI-MM1-samplerate` ha sido creado conteniendo los paths de cada fichero de audio. En este caso el contenido del `audiosource` sería:
  ```
  3 15
  /usr/share/geintra/.../LFD-P5-O1-U15004-ch1.wav
  /usr/share/geintra/.../LFD-P5-O1-U15004-ch2.wav
  /usr/share/geintra/.../LFD-P5-O1-U15004.wav
  /usr/share/geintra/.../LFD-P5-O1-U01171-ch1.wav
  /usr/share/geintra/.../LFD-P5-O1-U01171-ch2.wav
  /usr/share/geintra/.../LFD-P5-O1-U01171.wav
  ...
  ```
  Como se puede observar se van a emplear 4 micrófonos en el experimento, siendo necesario únicamente 3 archivos en cada seminario. Esto es debido a que el canal 3 y 4 del tercer fichero *wav* se obtiene de un archivo multicanal.
  A continuación se muestra otro ejemplo de uso de estos parámetros:
  ```
  SIZECOMMONFILENAME=15
  DB_ID="HIFI-MM1"
  DB_HOME_DIR="$HOME_DIR/corpora/speech-split/LFD"
  NUM_MICS=4
  NUMFILESPERUTT=2
  CHANNELS=( 1 2 3 )
  ```

  En este caso el contenido del fichero `audiosource` sería el siguiente:
  ```
  2 15
  /usr/share/geintra/.../LFD-P5-O1-U15004.wav
  /usr/share/geintra/.../LFD-P5-O1-U15004-ch4.wav ...
  ```

8.2.4.3. Tareas y configuración dependiente del experimento

1. Se lanza el algoritmo `SRP` con los parámetros correspondientes:

   - Se configuran los parámetros necesarios por el algoritmo `SRP` en el fichero de configuración `go-SRP.cfg`:
     ```
     FRAME_SIZE_SECS=0.32
     FFT_SIZE=16384
     FRAME_SHIFT_SECS=0.04
     SIM_FILENAME=.edecanRoom-HIFIMM1-srp.sim"
     CHANNELS="1,2,3"
     ...
     ```
     Es necesario recordar que el archivo de simulación debe estar localizado en el directorio del experimento. Ahora, se comprueba el contenio del fichero de simulación correspondiente al experimento.

   - Se ejecuta `go-SRP.sh`:
     `$ bash go-SRP.sh &`
     De esta manera la ejecución del programa se realizará en modo *background*. Cuando el proceso termine un email será recibido para notificarlo; en este email está disponible información sobre posibles errores. El formado del fichero `.log` es en este caso:
     `200902240957-HIFI-MM1-samplerate.log`.

8.2.4.4. Tareas y configuración dependiente de la evaluación

1. Se evalúan los resultados generados por el proceso de localización `SRP`.

   - Se confirman los parámetros correspondientes en `go-EVAL.cfg`. En este ejemplo:
     ```
     EVALAPP="$HOME_DIR/chilEvaluationSW/sp_loc_eval"
     OVERALLAPP="$HOME_DIR/chilEvaluationSW/calc_overall_performance.pl"
     ```

   - Se ejecuta `go-EVAL.sh`:
     `$ bash go-EVAL.sh`
     Los ficheros `.err` y `.log` se habrán creado.

   - Para evaluar sólo alguna secuencia de interés, se crea otro `.list`. Por ejemplo `P1.list` para evaluar sólo las secuencias del locutor P1. Se ejecuta entonces la aplicación de evaluación seguida por el nombre del nuevo fichero `.list`:
     `$bash go-EVAL.sh P1`
     El nombre del fichero `.err` y `.log` serían:
     `200902241014-HIFI-MM1-samplerate-P1-eval.log`
     `200902241014-HIFI-MM1-samplerate-P1-chilError.err`

8.2.5. Información adicional

A continuación se detallan algunos aspectos de interés a tener en cuenta.

8.2.5.1. Consideraciones a tener en cuenta para el correcto funcionamiento del sistema de experimentación

El sistema de experimentación funciona correctamente teniendo en cuenta varios aspectos:

- Los directorios `autoGenExp` y `experiments` deben encontrarse bajo el directorio `HOME_DIR` proporcionado en `genExp.cfg`.
- El fichero `.list` que contiene las secuencias a ser analizadas debe estar localizado en el directorio del experimento específico (nombrado de la misma manera que el identificados del experimento `DB_ID`), siendo necesario que el nombre del fichero `.list` sea `DB_ID.list`.
- En `go-SRP.cfg` existen ciertos parámetros que deben ser proporcionados obligatoriamente. Se debe prestar especial atención a sus valores.
- El fichero de simulación `.sim` debe encontrarse en el directorio del experimento.

8.2.5.2. ¿Cómo incluir una nueva base de datos en el sistema de experimentación?

Para hacer posible que el sistema de scripts funcione con una nueva base de datos, algunos cambios son necesarios:

1. En `go-GENDBLIS.cfg` el identificador de la base de datos `DB_ID` tendrá un nuevo posible valor, el mismo que el nombre de la nueva base de datos. El nombre está sujeto a elección por parte del usuario.
2. En `go-GENDBLIS.cfg`, `DB_HOME_DIR` debe ser completado con el path de los ficheros de audio de la nueva base de datos..
3. En `go-GENDBLIS.sh` existe un bucle `case` con todas las posibles bases de datos consideradas. Se debe añadir código en este bucle para generar el fichero `audiosource` con el formato específico de la nueva base de datos. Se debe poner atención si una línea es usada para construir los paths de todos los micrófonos pertenecientes a un array o únicamente a diferentes micrófonos. Compruebe en el `.list` que los resultados son los esperados.
4. Con la nueva base de datos debe existir el correspondiente fichero de simulación `.sim`. Si no existe, debe ser creado. Entonces, en `go-SRP.cfg` el campo `SIM_FILENAME` ha de ser actualizado.
5. Al igual que en `go-GENDBLIS.sh`, existe un bucle `case` en `go-EVAL.cfg` donde se debe añadir código relacionado con la nueva base de datos. El path de los ficheros de *groundtruths* debe ser construido en este bucle.
6. Finalmente, en `go-EVAL.sh` hay un nuevo bucle `case`. En esta ocasión, el path de los ficheros de *groundtruths* para cada archivo del `.list` se construye. Se debe prestar atención si para cada fichero del `.list` se debe crear más de un fichero *groundtruth*, es posible que se tengan que introducir algunos cambios. Por ejemplo en la base de datos ISL2004, cuatro *groundtruths* (cuatro micrófonos en cada array) son construidos para cada línea del `.list`.

8.3. Manual de usuario del sistema de experimentación basado en vídeo y procesamiento audiovisual

8.3.1. Introducción

El sistema de experimentación basado en vídeo y procesamiento audiovisual se construye empleando una arquitectura cliente-servidor, donde cada cámara (con información de audio o vídeo) es tratada como un servidor que envía información al cliente, siendo éste el encargado de procesar y mostrar dicha información.

En [54] se desarrolló la estructura software necesaria para lanzar servidores que envían imágenes capturadas en tiempo real y que son representadas por el cliente. Posteriormente se añadió la funcionalidad de trabajar en modo simulación, permitiendo así que los servidores envíen imágenes almacenadas en el disco duro del ordenador; siendo ya en este trabajado donde se han agregado el resto de funcionalidades mostradas en los siguientes apartados.

Es decir, la aplicación funcionaba con información procedente de visión y se le añade en este trabajo la posibilidad de trabajar con información acústica, planteando dos enfoques:

- Trabajar en tiempo real, empleando para tal los micrófonos que se encuentran en el Espacio Inteligente de la Universidad de Alcalá. De este modo se tiene una aplicación basada en fusión audiovisual capaz de localizar a las personas gracias a información procedente de cámaras y micrófonos en tiempo real en un espacio dado.
- Trabajar en modo simulación, empleando datos obtenidos de las bases de datos de CLEAR 2007 por ejemplo o de cualquier otra.

En la Figura 8.2 se observa la estructura del nuevo sistema de experimentación:

8.3.2. Descripción del sistema de experimentación

Como se puede observar en la imagen 8.3 el sistema de experimentación necesita una serie de datos de entrada y de salida que se explican a continuación:

- Argumentos del programa:
  - `calibration/homography file`: Se especifica el archivo que contiene los parámetros extrínsecos e intrínsecos de las cámaras (parámetros de calibración), o bien la matriz de homografía. En caso de proporcionar un archivo con parámetros de las cámaras, el sistema calcula automáticamente la matriz de homografía necesaria. Este parámetro es obligatorio.
  - `calib/homog used`: Se concreta si en el parámetro `calibration/homography` se ha definido un archivo con características de las cámaras (0) o con una matriz de homografía (1). Es un parámetro obligatorio.
  - `audio_involved`: Se detalla si el servidor envía imágenes procedentes del sistema de experimentación basado en audio (1) o son imágenes capturadas con cámaras (0). Éste es también parámetro obligatorio.
  - `audioPower file`: Este parámetro ha de suministrarse obligatoriamente en caso de emplear el sistema de experimentación de audio, especificando el nombre completo del fichero `.pow` donde se han almacenado las potencias para cada punto y cada instante de tiempo.

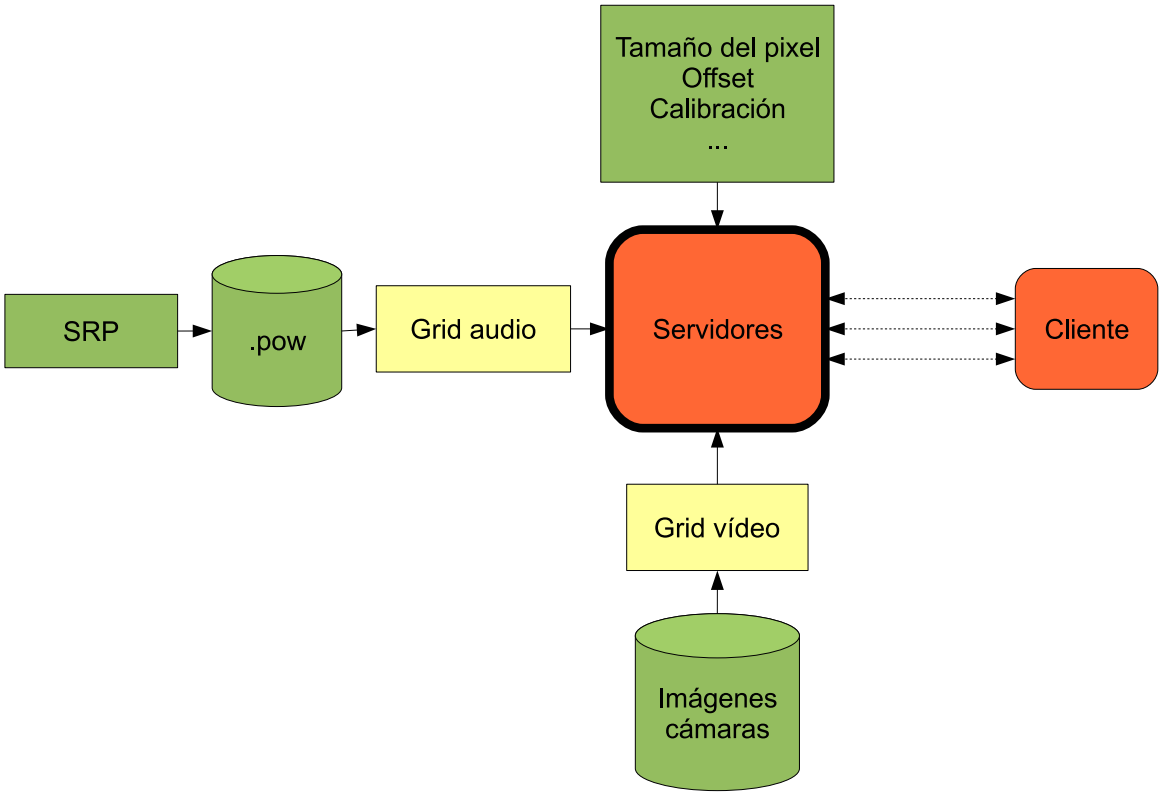


Figura 8.2: Estructura básica del sistema de experimentación basado en fusión audiovisual

- `grid_size_x`: Tamaño del píxel en la coordenada x expresado en milímetros, necesario en caso de haber proporcionado un archivo con parámetros de calibración de las cámaras para así obtener la correspondiente matriz de homografía.

- `grid_size_y`: Parámetro igual al anterior pero referido a la coordenada y. Como se ha explicado en la sección 5.4 de la página 74 el tamaño del píxel es un parámetro fundamental a la hora de calcular la matriz de homografía en el servidor, así como para representar gráficamente por parte del cliente. Para asignarle un valor apropiado al tamaño del píxel hay que tener en cuenta las dimensiones de la sala del experimento, con el fin de que se cubra todo el espacio de interés. Por ejemplo si se tiene una sala de dimensiones 5000 mm de ancho (coordenada x) y 3700 mm de alto (coordenada y) el tamaño del píxel ideal sería 16 (5000/320) y 16 (3700/240), teniendo en cuenta que la matriz de homografía se calcula siempre para una imagen de tamaño 320x240, tamaño en píxeles. Si se asigna un valor menor o mayor al idóneo a los parámetros de tamaño del píxel se calcula una matriz de homografía que cubre un espacio menor o mayor al deseado, y por lo tanto los resultados no son los deseados. En la Figura 8.4 se realiza el experimento con el tamaño apropiado para el tamaño del píxel y se comprueba que el espacio de representación (líneas verdes) coincide con las dimensiones de la sala. En la Figura 8.5 se ha asignado un tamaño menor al anterior y en la Figura 8.6 unos valores mayores a los óptimos, comprobando cómo los límites representados gráficamente no son los correctos, y de la misma manera aparecen errores en la representación de la clase validada (círculo).

Figura 8.3: Diagrama del sistema de experimentación

Figura 8.4: Experimento con tamaño del píxel de 16 mm

- `offset_homography_x`: Offset en la coordenada x respecto a la posición marcada como 0 empleada en el cálculo de la matriz de homografía, expresada en milímetros y obligatoria de proporcionar en caso de haber suministrado un archivo con parámetros intrínsecos y extrínsecos de la cámara. Todas las representaciones gráficas y la extracción de medidas de las imágenes de vídeo se llevan a cabo en la misma posición que la matriz de homografía.
- `offset_homography_y`: Parámetro idéntico al anterior pero referente a la coordenada y.
- `offset_homography_z`: Parámetro idéntico a los dos anteriores pero referente a la coordenada z. Estos tres parámetros permiten "desplazar" la matriz de homografía en las tres coordenadas. Esto es especialmente útil en la coordenada z ya que permite posicionar la matriz de homografía, y por tanto el espacio de búsqueda a una altura determinada. Por ejemplo las imágenes de audio se calculan para una altura distinta de 0 ya que la potencia de audio es mayor en alturas donde generalmente se encuentran las personas hablando (alrededor de 1700 mm si la persona se encuentre de pie). Por lo tanto, si se inlcuye en la experimentación un servidor de audio, los demás servidores de vídeo deben establecerse a la misma altura que él. En la Figura 8.7(a) se muestra una escena donde la matriz de homografía se calcula para la altura 0, es decir, en el

Figura 8.5: Experimento con tamaño del píxel de 10 mm

Figura 8.6: Experimento con tamaño del píxel de 22 mm

suelo. En la Figura 8.7(b) se observa la misma escena con una matriz de homografía calculada para una altura de 1700 mm.

- `server_offset`: Parámetro que permite lanzar varios servidores desde un mismo host, siendo útil para trabajar en modo simulación cuando se dispone de toda la información del experimento en una única máquina. Cada uno de los servidores lanzados desde un mismo host debe poseer un offset diferente, ya que éste está relacionado con el número de cámara de las imágenes que van a ser cargadas; en la sección 8.3.4 se muestra un ejemplo de esta funcionalidad. En el caso de lanzar servidores desde diferentes máquinas con diferentes direcciones IPs el número de offset es indiferente, pudiendo ser asignado cualquier valor.

- Datos de salida:

  - `xpfFileOutput.txt`: Fichero de salida generado por el cliente donde se muestra en una primera columna el instante de tiempo en segundos y en una segunda columna las posiciones en milímetros; en las coordenadas x,y,z; de los objetos o elementos validados por el Filtro de Partículas.

(a) (b)

Figura 8.7: Imágenes de un experimento con la matriz de homografía calculada para: (a) altura del suelo (b) altura de 1700 mm

8.3.3. HOWTO del sistema de experimentación

8.3.3.1. Configuración de los servidores

Para configurar correctamente cada uno de los servidores a emplear es necesario modificar y comprobar parámetros de ciertos ficheros fuente que se detallan a continuación:

- `ispace.h`: Fichero cabecera que contiene la declaración de constantes empleadas por la aplicación del servidor, siendo aquellas que hay que adaptar las siguientes:
  - `FRAMES_BACKGROUND`: Número de frames empleados por un servidor de vídeo (envía imágenes procedentes de las cámaras en tiempo real o imágenes almacenadas en el servidor) para calcular el fondo de la imagen.
  - `FRAMES_BACKGROUND_AUDIO`: El sistema emplea un método de validación del grid de audio, por el cual se calcula la desviación típica de los píxeles de una serie de frames iniciales, `FRAMES_BACKGROUND_AUDIO`.
  - `OFFSET_BACKGROUND`: Parámetro utilizado para especificar el nombre de la primera imagen a emplear en el cálculo del fondo, cargada cuando se trabaja en modo simulación. Por ejemplo si `OFFSET_BACKGROUND` es 130, la primera imagen leída será seq_000130.jpg.
  - `AUDIO_CALIB`: Parámetro que permite mostrar por pantalla la distribución estadística de la desviación típica de los píxeles, utilizado en el caso de que el servidor en cuestión envíe información de audio. Para ello se ejecuta en primer lugar un experimento únicamente en un intervalo de tiempo en el que haya silencio y se caracteriza la desviación típica con el fin de comprobar si ésta sigue una distribución Gaussiana y es correcto el método de validación del grid de audio empleado.
  - `SIMULACION`: Parámetro que indica si se está trabajando en modo simulación (1) o no (0).
  - `READREFERENCE`: En caso de ejecutar el sistema en modo simulación es posible visualizar en el cliente los *ground truths* sobre las imágenes, tanto los de vídeo como los de audio. De esta manera, será sencillo comprobar a simple vista el funcionamiento del programa.
  - `FPS`: Velocidad en frames por segundo de funcionamiento de los servidores de visión, pudiendo ser la velocidad a la que capturan las cámaras en caso de trabajar en tiempo

real o la velocidad a la que han sido grabadas las escenas de la base de datos empleada cuando se trabaja en modo simulación.

- `FPS_AUDIO`: Velocidad en frames por segundo de funcionamiento del servidor de audio. Si se trabaja en modo simulación es la velocidad a la que se han almacenado las muestras de potencia para las distintas localizaciones.
- `VIDEOROOTDIR`: Directorio raíz en el que se localizan los archivos `.ini` con los parámetros de calibración de las cámaras, los ficheros conteniendo los *ground truths* tanto del audio como del vídeo y las imágenes del seminario que van a ser enviadas por el servidor (tanto las de background empleadas para calcular el fondo como las propias del seminario).
- `SEMINAR`: Nombre del seminario sobre el que se va a llevar a cabo el experimento, únicamente aplicable en caso de trabajar en modo simulación.
- `VIDEODIR`: Nombre del directorio contenido dentro de `VIDEOROOTDIR/SEMINAR` en el que se localizan las imágenes propias del seminario a enviar por los servidores cuando se está realizando un experimento en modo simulación.
- `BCKDIR`: Nombre del directorio contenido dentro de `VIDEOROOTDIR/SEMINAR` en el que se localizan las imágenes empleadas para obtener el fondo cuando se está realizando un experimento en modo simulación.
- `INIDIRECTORY`: Nombre del directorio contenido dentro de `VIDEOROOTDIR/SEMINAR` en el que se localizan los ficheros con los parámetros extrínsecos e intrínsecos de las cámaras en caso de que los haya, indicándose en el parámetro de entrada `calib/homog used`.
- `POWER_FILE_DIRECTORY`: Nombre del directorio en el que se encuentra almacenado el fichero de potencias procedente del sistema de experimentación basado en audio, cuyo nombre se especifica en el parámetro de entrada `audioPower file`. Este parámetro es tenido en cuenta en caso de trabajar en simulación y lanzar un servidor de audio.
- `NUM_GRAY_SCALE`: Número de grises que se van a emplear para construir la imagen de audio a partir del fichero de potencias. Este parámetro es tenido en cuenta en caso de trabajar en simulación y lanzar un servidor de audio.

- `referencecfg.h`: Fichero donde se encuentran las constantes, estructuas y funciones empleadas en la representación gráfica de las posiciones etiquetadas (*ground truth*) en caso de tratarse del modo simulación.

  - `REF_DIRECTORY`: Nombre del directorio contenido en `VIDEOROOTDIR` donde se encuentran los archivos de *ground truth*.
  - `REF_FILE_AUDIO`: Nombre del fichero *ground truth* de audio.
  - `REF_FILE_VIDEO`: Nombre del fichero *ground truth* de vídeo.

- `red.c`: Contiene funciones relacionadas con la comunicación con el cliente.

  - Es necesario comprobar en la función `funcion_hilo_tx_frames` que se emplea el valor de frames por segundo adecuado en cada ocasión. Si todos los servidores son de vídeo se empleará el parámetro `FPS`, si sólo hay un servidor y es de audio se empleará `FPS_AUDIO`, y si existen servidores de audio y vídeo `FPS`.
  - En la función `enviar_frame` existen dos líneas que es necesario ajustar al experimento en cuestión, cuyas especificaciones de ajuste se encuentran en el propio código. Estas líneas son las siguientes:
    `if ((syncCont == FPS/FPS_AUDIO) && (numframe >FRAMES_BACKGROUND))`

```
if (numframe >FRAMES_BACKGROUND)
```

- `utils.c`: Conjunto de funciones de distinta naturaleza.
  - En la función `leer_frame_HDWOMalloc` es necesario adaptar y comprobar la construcción del nombre completo de las imágenes a enviar por el servidor en caso de simulación con servidores de vídeo. Un ejemplo de estos fragmentos a modificar es la siguiente construcción:
    `sprintf(cadena, VIDEOROOTDIR/"SEMINAR/"VIDEODIR/cam%d/seq_%.6d.jpg",offset_cam+1 ,frame-FRAMES_BACKGROUND);`

8.3.3.2. Lanzar los servidores

El ejecutable empleado para lanzar los servidores se denomina `gladeservi0` y se encuentra localizado dentro de la carpeta `src` en el directorio del servidor. La llamada a este ejecutable se muestra a continuación:

```
./gladeservi0 <calibration/homography file><calib/homog used><audio involved>[audioPower file] [grid_size_x] [grid_size_y] [offset_homography_x] [offset_homography_y] [offset_homography_z] [server_offset]
```

En la seccion 8.3.4 se muestran varios ejemplos de llamadas a `gladeservi0` para lanzar servidores que envían información procedente del audio o de visión.

8.3.3.3. Configuración del cliente

Con el fin de configurar debidamente la aplicación cliente es necesario comprobar una serie de parámetros que se explican a continuación:

- `constantes.h`: Valor de constantes empleadas por el cliente.
  - `GRIDSIZE_X/GRIDSIZE_Y`: Tamaño de cada píxel en milímetros empleado por el cliente para las coordenadas x e y, que debe ser elegido intentando que éste sirva para cubrir el espacio total de la habitación donde se va a llevar a cabo el experimento, teniendo en cuenta que la imagen del cliente tiene un tamaño de 320x240 píxeles. En caso de haber especificado en la llamada al servidor el tamaño del píxel (parámetro `grid_size_x, grid_size_y`) éste debería tener el mismo valor.
  - `GRIDSTEP`: Existe la posibilidad de saltarse píxeles a la hora de obtener las medidas a partir del grid final con el fin de reducir el número de las mismas y disminuir el tiempo de cómputo. Este parámetro indica el salto en píxeles y debe ser múltiplo de 3 (el color del píxel se identifica por 3 valores: RGB).
  - `REFRESCO_ZONAS_DIBUJO`: Tiempo en milisegundos empleado por el cliente para llevar a cabo el refresco de las imágenes, que debe ser igual o menor al valor correspondiente teniendo en cuenta los frames por segundo del experimento. Es decir, si un experimento se realiza a 5 frames por segundo, el valor máximo de la constante debería ser 200 milisegundos para visualizar todos los frames enviados por el servidor.
  - `TINIT`: En caso de escribir el fichero de salida `xpfFileOutput.txt` es necesario grabar los instantes de tiempo con un formato igual al que se encuentra en los ficheros de

ground truth del experimento, para posteriormente poder llevar a cabo una evaluación. En algunos casos el instante inicial no es 0, sino otro valor que debe ser proporcionado en `TINIT`. En caso de que el instante inicial sea 0 es necesario darle un valor 0 a la constante `TINIT`.

- `miXpfcp.h`: Fichero que contiene constantes empleadas por el Filtro de Partículas.
  - `PRINTRESULTS`: Si se desea grabar los resultados del Filtro de Partículas en el fichero `xpfFileOutput.txt` hay que establecer esta variable a 1, en caso contrario se le asigna un valor 0.

8.3.3.4. Lanzar el cliente

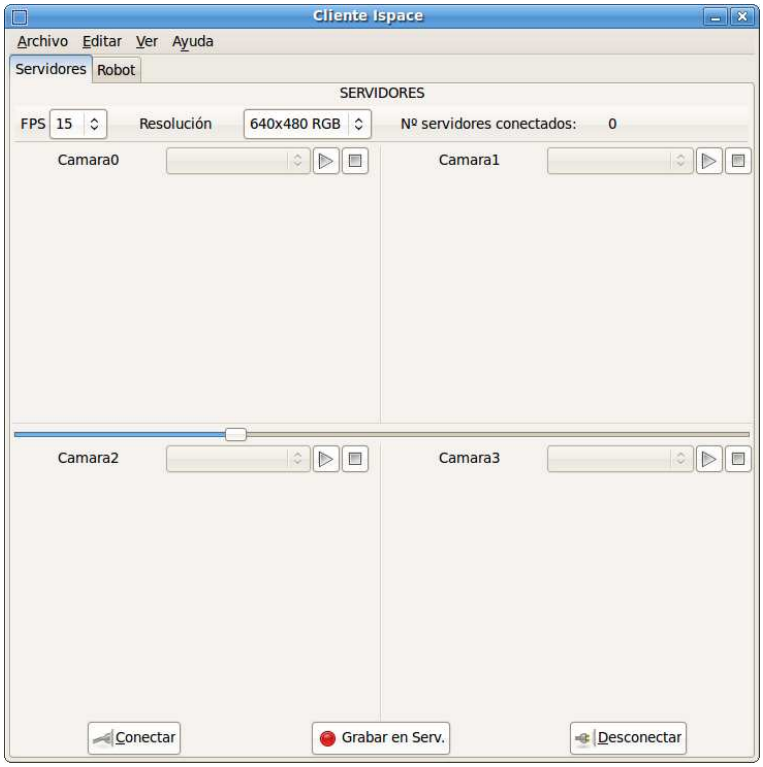


Figura 8.8: Aplicación del cliente

Para ejecutar el cliente se debe hacer la siguiente llamada dentro del directorio `src` del cliente: `./proyecto2`.
En la figura 8.8 se puede ver la pantalla de la aplicación cliente, donde se van a comentar únicamente aquellas utilidades de interés en este trabajo. Para más información consultar [54].

En primer lugar es necesario definir los servidores que van a ser conectados, accediendo al menú *Editar* y posteriormente *Servidores*. Aparece entonces una ventana como la mostrada en la figura 8.9. En este menú se observan una serie de servidores predefinidos (Servidores Ispace) correspondientes a aquellos conectados a las cámaras localizadas en el Espacio Inteligente del Departamento de Electrónica de la Universidad de Alcalá. Además aparecen otros servidores, pudiendo ser configurados por el usuario, añadiendo, eliminando, guardando como predeterminado, etc. Es necesario aclarar que el formato de la dirección de los servidores es:
`<dirección IP>:<offset>`

Figura 8.9: Menú de edición de los servidores

Siendo este offset el indicado en la llamada a la aplicación del servidor mediante el parámetro `server_offset`.

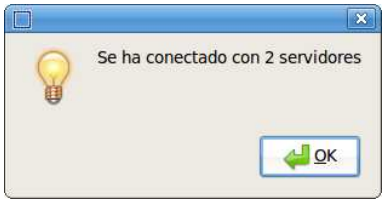


Figura 8.10: Conexión realizada con éxito

En esta ventana hay que seleccionar aquellos servidores que se desean conectar y pulsar el botón *Aceptar*. Posteriormente se puede pulsar *Conectar* para comenzar el envío de información, apareciendo por pantalla una ventana como la que se puede observar en la figura 8.10 en caso de que la conexión se haya realizado satisfactoriamente. Una vez conectados, se puede seleccionar en cada una de las cuatro ventanas del cliente las imágenes que se desean visualizar, pudiendo ser aquellas enviadas por los servidores o la composición del grid.

8.3.4. Ejemplo

Se va a mostrar un experimento completo donde se emplea un servidor de audio y dos de vídeo, trabajando en modo simulación. El experimento en concreto corresponde a la sala AIT de CLEAR 2007 (seminario AIT_20060728).

8.3.4.1. Configurar los parámetros en servidor y cliente

Cuando se va a realizar un experimento nuevo es necesario comprobar y modificar el valor de varios parámetros que ya han sido mencionados y explicados con anterioridad. A continuación se muestran todos ellos así como los ficheros en los que se encuentran:

Servidor:

- `ispace.h`. Hay que comprobar los siguientes parámetros que en este ejemplo adquieren estos valores:

```
#define FRAMES_BACKGROUND 4
#define FRAMES_BACKGROUND_AUDIO 87
#define OFFSET_BACKGROUND 0
#define AUDIO_CALIB 0
#define SIMULACION 1
#define READREFERENCE 1
#define FPS 30
#define FPS_AUDIO 10
#define VIDEOROOTDIR /usr/share/geintra/databases/mmodal/CHIL_D7_14/CHIL_D7_14/DATA
/DEV/SEMINARS"
#define SEMINAR ´´AIT_20060728"
#define VIDEODIR "video"
#define BCKDIR "info/background"
#define INIDIRECTORY "info/calibration"
#define POWER_FILE_DIRECTORY /home/maria/ait"
#define NUM_GRAY_SCALE 256
```

- `referencecfg.h`. Es necesario asignar un correcto valor a los siguientes parámetros relacionados con el seminario:

```
#define REF_DIRECTORY "video_labels"
#define REF_FILE_AUDIO ´´AIT_20060728_Acoustic3d_label.txt"
#define REF_FILE_VIDEO ´´AIT_20060728_3d_label.txt"
```

- `red.c`. Al estar implicados en el experimento servidores de audio y vídeo, la función `funcion_hilo_tx_frames` aparece de la siguiente manera:

```
void  funcion_hilo_tx_frames(void arg)
{
    while(mantener_hilo_tx_frame)
    {
       usleep((int)(1000000/FPS));
       if(trabajando_en_tx_frame)
       ...
}
```

En el caso de intervenir únicamente servidores de vídeo no se produce ningún cambio en esta función. Si sólo se empleara en el experimento un único servidor de audio, la función quedaría como sigue, habiendo cambiado el tiempo de espera en el envío de imágenes al cliente:

```
void  funcion_hilo_tx_frames(void arg)
{
    while(mantener_hilo_tx_frame)
    {
       usleep((int)(1000000/FPS_AUDIO));
       if(trabajando_en_tx_frame)
       ...
}
```

Además, en la función `enviar_frame` hay que comprobar dos líneas de código que en este caso se establecen como se ve a continuación:

```
if ((syncCont == FPS/FPS_AUDIO) && (numframe >FRAMES_BACKGROUND))
if (numframe >FRAMES_BACKGROUND)
```

Si sólo se empleara en el experimento un servidor de audio, las mismas líneas se configurarían así:

```
if (syncCont == 1)
//if (numframe >FRAMES_BACKGROUND)
```

Por último, en el caso de realizar un experimento con servidores de vídeo:

```
if ((syncCont == 1) && (numframe >FRAMES_BACKGROUND))
if (numframe >FRAMES_BACKGROUND)
```

- `utils.c`. En la función `leer_frame_HDWOMalloc` hay que adaptar las líneas correspondientes a la carga de las imágenes de la base de datos para los servidores de vídeo, siendo las siguientes para el presente experimento:

```
if (frame <FRAMES_BACKGROUND)
    sprintf(cadena, VIDEOROOTDIR/"SEMINAR/"BCKDIR/cam%d_bkg%d.jpg",offset_cam+1,
    (((frame+OFFSET_BACKGROUND)%3)+1))
if (frame == FRAMES_BACKGROUND)
{
    end_background = 1;
}
if (end_background)
    sprintf(cadena, VIDEOROOTDIR/"SEMINAR/"VIDEODIR/cam%d/seq_%.6d.jpg",offset_cam+1,
    frame-FRAMES_BACKGROUND)
...
```

Cliente:

En la aplicación del cliente también es necesario adaptar ciertos parámetros al experimento en cuestión. A continuación se muestra la configuración de los mismos para el experimento AIT de CLEAR 2007:

- `constantes.h`. Es necesario testear el valor de las siguientes constantes:
  `#define GRIDSIZE_X 16`

```
#define GRIDSIZE_Y 16
#define GRIDSTEP 9
#define REFRESCO_ZONAS_DIBUJO 33
#define TINIT 1150792357.97
```

- `miXpfcp.h` Si se quiere obtener un fichero de salida se comprueba el valor del siguiente parámetro:
  `#define PRINTRESULTS 1`

8.3.4.2. Lanzar los servidores

A continuación se muestran varios ejemplos de llamadas a la aplicación servidor para los distintos tipos de experimentos:

- Servidor de audio:
  `./gladeservi0 homographyAUDIO.ini 1 1 AIT_20060728.pow 0`
- Uno o varios servidores de vídeo:
  `./gladeservi0 cam1.ini 0 0 16 16 0 0 1700 0`
  `./gladeservi0 cam2.ini 0 0 16 16 0 0 1700 1`
  `...`
- Un servidor de audio y uno o varios servidores de vídeo:
  `./gladeservi0 homographyAUDIO.ini 1 1 AIT_20060728.pow 0`
  `./gladeservi0 cam2.ini 0 0 16 16 0 0 1700 1`
  `...`
  Obsérvese que en estos casos el parámetro `server_offset` del servidor de audio siempre debe ser 0.

Como ejemplo se va a tomar el último caso al ser el más completo ya que intervienen servidores de vídeo y uno de audio.

En el momento en que se lanza cada uno de los servidores, aparece por pantalla cierta información de interés, siendo ligeramente diferente en el caso de tratarse de un servidor de audio o vídeo:

- Servidor de audio:
  En primer lugar aparece información sobre el fichero del que se obtiene la matriz de homografía, si éste contiene directamente una matriz de homografía o parámetros de calibración, si el servidor es de audio o vídeo, y en caso de que sea de audio se indica el fichero del que se leen los valores de potencia.

  ```
  Initialize file: homographyAUDIO.ini
  homographyFile: 1
  Audio involved: 1
  Power file: /home/maria/ait/xaa
  ```

  A continuación se observa por pantalla la distribución en los valores de potencia del experimento, con el fin de que el usuario pueda establecer el rango óptimo para la umbralización de la imagen explicada en la sección 5.5.2 de la página 78. No se muestra esta distribución ya que son multitud de datos.

Tras la distribución de potencias aparecen otros datos de interés como son el offset que indica el número del servidor cuando se lanzan varios desde un mismo host, la matriz de homografía, el nombre del equipo, la IP, el fichero desde el que se leen las posiciones etiquetadas y los sockets creados para comunicarse con el cliente:

```
Offset: 0
homographyMatrix->H[0][0]=1.000000
homographyMatrix->H[0][1]=0.000000
homographyMatrix->H[0][2]=0.000000
homographyMatrix->H[1][0]=0.000000
homographyMatrix->H[1][1]=1.000000
homographyMatrix->H[1][2]=0.000000
homographyMatrix->H[2][0]=0.000000
homographyMatrix->H[2][1]=0.000000
homographyMatrix->H[2][2]=1.000000
maria-laptop
IP: 127.0.1.1:0
Reading /usr/share/geintra/databases/mmodal/CHIL_D7_14/CHIL_D7_14/DATA/DEV/SEMINARS/
video_labels/AIT_20060728/AIT_20060728_Acoustic3d_label.txt file
Socket created address:127.0.0.1 port:1500
Socket binded address:127.0.0.1 port:1500
Socket created address:127.0.0.1 port:1520
Socket binded address:127.0.0.1 port:1520
Socket created address:127.0.0.1 port:1490
Socket binded address:127.0.0.1 port:1490
```

- Servidor de vídeo:
  Se observa en la pantalla información sobre el nombre del fichero a partir del cual se calcula la matriz de homografía, si éste contiene una matriz o parámetros de calibración, si es un servidor de audio o no, el tamaño del píxel en mm, el offset para el cálculo de la matriz de homografía, el offset que indica el número del servidor cuando se lanzan varios desde un mismo host, la matriz de homografía calculada, el nombre del equipo, la IP, el fichero del que se obtienen las posiciones etiquetadas y los sockets creados para la comunicación con el cliente.

```
Initialize file: cam2.ini
homographyFile: 0
Audio involved: 0
Grid size x: 16.000000, gride size y: 16.000000
Homography offset x: 0.000000, homography offset y: 0.000000, homography off-
set z: 0.000000
Offset: 1
H[0]=-0.000043
H[1]=-0.000017
H[2]=0.069498
H[3]=-0.000074
H[4]=0.000130
H[5]=-0.007446
H[6]=-0.000000
```

```
H[7]=0.000001
H[8]=0.000024
maria-laptop
IP: 127.0.1.1:1
Reading /usr/share/geintra/databases/mmodal/CHIL_D7_14/CHIL_D7_14/DATA/DEV/SEMINARS/
video_labels/AIT_20060728/AIT_20060728_3d_label.txt file
Socket created address:127.0.0.1 port:1501
Socket binded address:127.0.0.1 port:1501
Socket created address:127.0.0.1 port:1521
Socket binded address:127.0.0.1 port:1521
Socket created address:127.0.0.1 port:1491
Socket binded address:127.0.0.1 port:1491
```

En este momento los servidores están esperando a recibir comandos del cliente para comenzar las capturas, por lo que se está en disposición de lanzar la aplicación cliente.

8.3.4.3. Lanzar el cliente

En primer lugar para comenzar la ejecución del cliente hay que escribir el comando correspondiente en el directorio del cliente:
`./proyecto2`

En este momento aparece por pantalla una ventana como la de la Figura 8.8. En ella abrimos el menú Editar->Servidores y seleccionamos en nuestro ejemplo tres de ellos, con la IP que se ha observado al lanzar los servidores y el offset correcto. Esta selección se muestra en la Figura 8.11:

Posteriormente se pulsa *Aceptar*, y en la ventana principal se procede a darle al botón *Conectar* que se puede ver en la Figura 8.8. Si la conexión se ha realizado con éxito debe aparecer el mensaje que se observa en la Figura 8.12.

En este momento comienza la comunicación entre el cliente y servidor. Para visualizar las imágenes de cada servidor y del grid se selecciona en cada una de las ventanas la imagen que se desea ver. Para ello se pulsa el botón sobre el que se encuentra el puntero en la Figura 8.13. Aparece una lista desplegable con todos los servidores conectados y el grid para escoger uno de ellos. Después, si se pulsa el botón Play marcado con el puntero del ratón en la Figura 8.14 comienza la visualización sobre esa ventana. Se procede de la misma manera con las cuatro ventanas disponibles.

Para más detalles sobre todas las posibilidades ofrecidas por esta aplicación ver el trabajo realizado en [54].

8.3.5. Información adicional

8.3.5.1. Consideraciones a tener en cuenta para el correcto funcionamiento del sistema

Para conseguir el funcionamiento esperado hay que tener en cuenta ciertas consideraciones:

- Existen una serie de parámetros obligatorios que deben ser especificados por el usuario y pasados como entrada a la aplicación del servidor `gladeservi0`.

Figura 8.11: Selección de los servidores

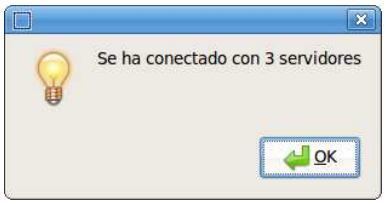


Figura 8.12: Conexión realizada con éxito con tres servidores

Figura 8.13: Forma de seleccionar un servidor

- En caso de que se vaya a lanzar un servidor de audio, el parámetro `server_offset` de éste debe ser 0. Si además existiera un servidor de vídeo, al ser su `server_offset` mayor o igual que 1, la primera cámara que se podría tener en cuenta en el experimento sería la cámara 2 (el número de la cámara se calcula a partir de dicho parámetro).

- Si se proporcionan ficheros `.ini` con la matriz de homografía en el parámetro `calibra-`

Figura 8.14: Forma de comenzar la visualización

`tion/homography file` éstos deben estar localizados en el directorio `src` del servidor.

8.4. Conclusiones

Se han descrito los manuales de usuarios detallados del sistema de localización acústica y el sistema de localización basado en fusión audiovisual, donde se muestran una a una las fases y pasos que hay que seguir para lanzar experimentos con éxito. Gracias a esto es posible realizar un experimento con facilidad y replicar los mostrados en este libro.

Capítulo 9

Apéndices

9.1. CHIL 2005, 2006 and 2007 Evaluation Packages: Summary of datasets for audio+visual person tracking

9.1.1. Introduction

In the context of the audio and video based speaker localization system available in the Geintra Research Group, this document shows all the needed data of the CHIL 2005, 2006 and 2007 evaluation packages distributed by ELRA, to be used in audio+video+multimodal person tracking. Information related to available audio and video files, calibration data, format of ground truth files and results is shown.

9.1.2. Context: CHIL project and CLEAR evaluations

9.1.2.1. CHIL project

CHIL - Computers in the Human Interaction Loop - is an Integrated Project under the European Commission's Sixth Framework Programme. It is jointly coordinated by Universitat Karlsruhe (TH) and the Fraunhofer Institute IITB. The project was launched on January, 1st 2004 and has a duration of 44 months.

The objective of this project is to create environments in which computers serve humans who focus on interacting with other humans as opposed to having to attend to and being preoccupied with the machines themselves. Instead of computers operating in an isolated manner, and Humans [thrust] in the loop [of computers] they will put Computers in the Human Interaction Loop (CHIL). They design Computer Services that model humans and the state of their activities and intentions. Based on the understanding of the human perceptual context, CHIL computers are enabled to provide helpful assistance implicitly, requiring a minimum of human attention or interruptions.

For further information on CHIL refer to the project web site at http://chil.server.de.

9.1.2.2. CLEAR evaluation

The CLEAR evaluation and workshop is an international effort to evaluate systems that are designed to recognize events, activities, and their relationships in interaction scenarios. CLEAR is meant to bring together projects and researchers working on related technologies in order to establish a common international evaluation in this field. The CLEAR evaluations are supported by the European Integrated project CHIL and the US National Institute of Standards and Technology (NIST).

9.1.3. 2005 CHIL Evaluation Campaign

9.1.3.1. Introduction

This evaluation campaign consists of two different databases regarding person tracking:

- ISL Seminar'03
- ISL Seminar'04

During this campaign, different technological components were evaluated:

- Vision technologies
  - Face Detection
  - Visual Person Tracking
  - Visual Speaker Identification
  - Head Pose Estimation
  - Hand Tracking
- Sound and Speech Technologies
  - Close-Talking Automatic Speech Recognition
  - Far-Field Automatic Speech Recognition
  - Acoustic Person Tracking
  - Acoustic Speaker Identification
  - Speech Activity Detection
  - Acoustic Scene Analysis

Next, a summary of the content for every evaluation package is shown:

- Vision technologies
  - Face Detection: The goal in this evaluation process is to assess the quality and accuracy of the face detection (and tracking) techniques being used and developed within the context of CHIL.
  - Visual Person Tracking: The task in this evaluation is to find the position of a speaker giving a presentation in front of an audience in the CHIL room. The speaker position is determined by the 3D room coordinates of the speakerś headś centroid.
  - Visual Speaker Identification: The evaluation protocol encompasses the following scenario:
    - Training: Frontal still images (manual operations allowed).
    - Testing: Multi-view video sequences (automatic)
  - Head Pose Estimation: The goal of this evaluation is to estimate a personś head pose. The head pose is determined by 3 angles: roll, pitch (also called tilt) and yaw (also called pan).
  - Hand Tracking: This evaluation defined a unique task, which is to find the positions of a personś left and right hand in all frames of a video sequence. The hand position is determined by the image coordinates (resp. 3D world coordinates) of the hand centroid.
- Sound and Speech Technologies
  - Close-Talking Automatic Speech Recognition: For training, due to the small size of the development sets of the ISL Seminar ́03 and Seminar ́04 sets (see below), the participating evaluation sites were free to use any type of speech corpora. In addition, the development sets of the two ISL Seminar databases were allowed for training (adaptation) of the acoustic and language models.

- Far-Field Automatic Speech Recognition: All training and supervised adaptation were to consider the development set as a whole, i.e., per-speaker supervised adaptation was not allowed. Unsupervised adaptation, per-speaker, was allowed on the test set.
- Acoustic Person Tracking: Training and development material was, in general, used to tune the speaker localization systems and in some cases limit the area where the source can be located. Metrics are based on the preliminary availability of reference files reporting the 3D true coordinates of the acoustic source (e.g. speaker), reported frame by frame (i.e. every 667 ms).
- Acoustic Speaker Identification: Two tasks were included in this evaluation run, a Speaker Identification (SI) task and a Speaker Verification (SV) task. For SI, Systems will have to give an identity to each test segment The SV task is to decide whether or not a specified speaker is speaking during a given speech segment.
- Speech Activity Detection: SAD systems can be trained by the portion of the database designated for that purpose. Each participant in the SAD evaluation is responsible to produce the output of its SAD system in the defined SAD XML format.
- Acoustic Scene Analysis: Acoustic events have been split into semantic classes and acoustic classes.
  - Semantic classes are: applause, breath, chair, click, door, footsteps, laugh, mouth noise, paper, sil, speech, throat, typing, unk, whir.
  - Acoustic classes are: c-t, c-nt, nc-s, nc-xreg, nc-xirr, n.

All our experiments are focused on Acoustic and Visual Person Tracking.

There is only one possible scenario, the seminar scenario, where the objective is to track a presenter talking to an audience in a seminar.

9.1.3.2. ISL Seminar 2003

This database contains audio and video recordings of 7 seminars that took place and were recorded at ISL.

Availabe data Concerning AUDIO ,the whole recordings of these seminars are not used for evaluation purposes: for each seminar, only 4 segments of 5 minutes are used. For each of these 4 segments, the audio files recorded from the lecturer's close talking microphone (CTM), and from channel 0 of a NIST MarkIII far-field microphone array are provided. The used algorithm SRP needs, at least, two audio signals coming from one or several static microphones. For that reason, we don't use this seminar in the experimentation related to audio person tracking, but it could be possible use the available video data, thus next they are detailed.

Concerning VIDEO, the recordings were made using 4 fixed SONY DFW-V500 640x480 colour firewire cameras in the room corners, at about 2.70m height and one pan-tilt-zoomable Canon VC-C1 640x480 colour analog camera at the far end of the room, opposite to the presenter area, to capture close ups of the speaker. The recorders were made at 15 frames per second. The captured RGB images are 640x480, 3 byte depth, and are supplied in the form of single jpeg files.

The available data belong to the 4 segments of the 7 seminars as explained before. Thus, in the directory `CHIL_D7_6/DATA/ISL_Seminar2003/VIDEO`.

The 7 seminars recorded in this database are:

- Seminar_2003-10-28
- Seminar_2003-11-11
- Seminar_2003-11-18
- Seminar_2003-11-25_A
- Seminar_2003-11-25_B
- Seminar_2003-12-16_A
- Seminar_2003-12-16_B

Next, the number of frames of each segment is shown:

Seminar	Segments	Frames
Seminar_2003-10-28	1	4279
	2	3994
	3	4268
	4	4278
Seminar_2003-11-11	1	4258
	2	4190
	3	4265
	4	4259
Seminar_2003-11-18	1	4274
	2	4278
	3	4256
	4	4274
Seminar_2003-11-25_A	1	4258
	2	4267
	3	4247
	4	4236
Seminar_2003-11-25_B	1	4270
	2	4236
	3	3104
	4	4252
Seminar_2003-12-16_A	1	4490
	2	4493
	3	4489
	4	4491
Seminar_2003-12-16_B	1	4494
	2	4483
	3	4494
	4	4492

Inside the directory of each seminar, there is a `.tar` file containing the `.jpg` of the 4 cameras.

Ground truth Video annotations were realised using an ad hoc tool provided by Universitat Karlsruhe. This tool allowed displaying 1 over 10 pictures in sequence, for the 4 cameras. On each displayed picture, the annotator had to click on the lecturerś head centroid, i.e. the estimated centre of his/her head if his/her face was visible. The 2D coordinates of the hit point within the camera plane were saved to the corresponding camn_label.txt file for further interpolation among all cameras in order to compute the real "ground truth"location of the speaker within the room.

The centroid of the speakerś head is provided in the form 3D coordinates (in mm relatives to the upper left corner of the checkerboard).

Then, the 3D ground truths of the video files are located in the directory
`CHIL_D7_6/TECHNOLOGIES/VIDEO/VISUAL_PERSON_TRACKING/REFERENCE`,
where a subdirectory of each segment exists. Inside the corresponding subdirectory a file named `3d_label.txt` is located. In this reference file the times are computed as absolute time in seconds from 1.1.1970 and the coordinates are in mm.

Furthermore, the groundtruth related to each camera is provided and named `camX_label.txt`. In this file the coordinates are in pixels and about one label per second has been labeled. These references are in 2D.

Calibration data Besides, the cameras calibration data are supplied, and they are located in the directory:
`CHIL_D7_6/TECHNOLOGIES/VIDEO/VISUAL_PERSON_TRACKING/TOOLS/calib/2003`.

Available results In the directory `CHIL_D7_6/TECHNOLOGIES/VIDEO/VISUAL_PERSON_TRACKING/RESULTS` an overview of the results obtained is shown.

9.1.3.3. ISL Seminar 2004

Available data This database contains audio and video recordings of 5 seminars:

- Seminar_11_11_A
- Seminar_11_11_B
- Seminar_11_11_C
- Seminar_11_12_A
- Seminar_11_12_B

All the available data are located in `CHIL_D7_6/DATA/ISL_Seminar2004`.

Concerning AUDIO, a certain number of segments are provided , at least 2 segments to be used for test purpose and called E1 and E2, and 2 segments for training/development purpose called D1 and D2 (if any). These segments are of approximately 15 min.

For each segment, a downsampled (16 KHz, 2 bytes) WAV file is provided for CTM and for channel #4 of MarkIII, along with original recordings:

- the CTM channel for the main speaker (44 KHz, 4 bytes WAV)

- a 64-channels Sphere file from the corrected NIST MarkIII microphone array, recording from the back of the room.

- 4 T-shaped arrays of 4 channels each recording from the surrounding walls.

- 4 single-channel tabletop microphones on the central table.

- a central single-channel on the central table.

Concerning VIDEO, the recordings were made using colour firewire cameras, at 15 frames per second. The captured RGB images are 640x480, 3 byte depth, and are supplied in the form of single jpeg files. As the cameras were only loosely synchronized, for each camera a separate index file is supplied listing the exact capture time for each frame. These timestamps could be used if a more precise synchronisation is required.

4 segments are given (from 1 to 4). The 2 first segments are for testing while the 2 last ones are for development purposes. In addition , up to 4 other segments may be given for each seminar (from A to D). They are provided to make more data available for certain technological components, so only the reference files of these segments are supplied.

Next, the available segments of audio and video technology are shown:

Seminar	Audio	Video
Seminar_11_11_A	E1 E2	1 2 3 4
Seminar_11_11_B	E1 E2 D1	1 2 3 4
Seminar_11_11_C	E1 E2 D1 D2	1 2 3 4
Seminar_11_12_A	E1 E2	1 2 3 4
Seminar_11_12_B	E1 E2	1 2 3 4

Following, the mapping from audio segments to video segments is as follows:

Seminar	Audio	Video
Seminar_11_11_A	E1	1 3 A
	E2	2 4
Seminar_11_11_B	E1	A 3 1
	E2	C
	D1	B 4 2
Seminar_11_11_C	E1	A 1 3
	E2	C 4 2
	D1	B
	D2	D
Seminar_11_12_A	E1	1 3
	E2	4 2
Seminar_11_12_B	E1	1 3 A
	E2	B 4 2

Next, the number of frames of each segment is shown:

Seminar	Segments	Frames
Seminar_11_11_A	1	4605
	2	4559
	3	4502
	4	4490
Seminar_11_11_B	1	6133
	2	5366
	3	4502
	4	4503
Seminar_11_11_C	1	4502
	2	5450
	3	4730
	4	4502
Seminar_11_12_A	1	3437
	2	5100
	3	4096
	4	2958
Seminar_11_12_B	1	4266
	2	5187
	3	4503
	4	4502

Ground truth For all video sequences the centroids labels are supplied and a rectangles following the face of the speaker, for all 4 cameras. These faces rectangles are provided for segments 1 to 4 only and are obtained with a specific tool. The centroid of the speaker's head is provided in the form 3D coordinates (in mm relatives to the north-west room center, referred to the floor). A 3D position label is provided for the timepoints corresponding to every 10th frame in camera sequence 1 (master camera), and only if the head is visible by at least two cameras at this timepoint. If the head is invisible or only visible by one camera, a (-1, -1, -1) label is provided.

Then, the 3D ground truths of the video files are located in the directory `CHIL_D7_6/TECHNOLOGIES/VIDEO/VISUAL_PERSON_TRACKING/REFERENCE`, where a subdirectory of each segment exists. Also you can find these ground truths in the directory `CHIL_D7_6/DATA/<Seminar>/VIDEO/<Segment>/LABELS_<Seminar>`. Inside the corresponding subdirectory a file named `3d_label.txt` is located. In this reference file the times are computed as absolute time in seconds from 1.1.1970 and the coordinates are in mm.

Furthermore, the groundtruth related to each camera is provided and named `camX_label.txt`. In this file the coordinates are in pixels and about one label per second has been labeled. These references are in 2D.

The groundtruths of the audio files are in the directory `CHIL_D7_6/TECHNOLOGIES/AUDIO/ACOUSTIC_PERSON_TRACKING/REFERENCE`. The times are relative times, start recording time is 0.

In the `.sim` file there is a variable named `fileFormat` to specify the ground truth files format. In this seminar its value is `CHIL_ACOUSTIC_PERSON_TRACKING_REFERENCE2006`.

Although the times in audio and video reference files are different , both are related to the same instants of time.

Next, the name of each file are detailed:

Seminar	Segments	Reference
Seminar_11_11_A	E1	Ref_MKIII_2004_11_11_1100_E1
	E2	Ref_MKIII_2004_11_11_1100_E2
Seminar_11_11_B	E1	Ref_MKIII_2004_11_11_1400_E1
	E2	Ref_MKIII_2004_11_11_1400_E2
	D1	Ref_MKIII_2004_11_11_1400_D1
Seminar_11_11_C	E1	Ref_MKIII_2004_11_11_1545_E1
	E2	Ref_MKIII_2004_11_11_1545_E2
	D1	Ref_MKIII_2004_11_11_1545_D1
	D2	Ref_MKIII_2004_11_11_1545_D2
Seminar_11_12_A	E1	Ref_MKIII_2004_11_12_1030_E1
	E2	Ref_MKIII_2004_11_12_1030_E2
Seminar_11_12_B	E1	Ref_MKIII_2004_11_12_1400_E1
	E2	Ref_MKIII_2004_11_12_1400_E2

The audio files have been evaluated every 100 ms and the reference are reported frame by frame (every 667 ms).

Calibration data Besides, the cameras calibration data are supplied, and they are located in the directory:
`TECHNOLOGIES/VIDEO/VISUAL_PERSON_TRACKING/TOOLS/calib/2004`.

Available results In the directory `TECHNOLOGIES/VIDEO/VISUAL_PERSON_TRACKING/RESULTS` an overview of the results obtained is shown.

9.1.4. 2006 CHIL Evaluation Campaign

9.1.4.1. Introduction

In this campaign seminars were recorded at 5 sites:

- UKA-ISL
- ITC-IRST
- AIT-RESIT
- UPC
- IBM

During this campaign, different technological components were evaluated:

- Vision technologies
  - Face and Head Detection

  - Visual Person Tracking
  - Visual Speaker Identification
  - Head Pose Estimation
- Audio Technologies
  - Acoustic Person Tracking
  - Acoustic Speaker Identification
- Multimodal Technologies
  - Multimodal Person Tracking
  - Multimodal Person Identification

Next, a summary of the content for every evaluation package is shown:

- Vision technologies
  - Face and Head Detection: The goal of this evaluation process is to assess the quality and accuracy of the face/head detection and tracking techniques being used and developed within the context of CHIL.
  - Visual Person Tracking: The objective is to track people as a whole, but for convenience of evaluation, a personś position is defined as the coordinates of the centre of his or her head, projected to the ground. Person tracking can be used as a building block for many other tasks such as face detection, identification, head pose estimation, focus of attention, etc
  - Visual Speaker Identification: The goal of this task it to recognize a person using short image sequences (up to 20 seconds). The evaluation was performed on two different databases,a seminar database and an interactive seminar database. On the seminar database the task is to identify only the presenter, whereas on the interactive seminar database, the presenter and the participants of the interactive seminar have to be identified.
  - Head Pose Estimation: The goal of this evaluation task is to estimate a personś head pose. The head pose is determined by 3 angles: roll, pitch (also called tilt) and yaw (also called pan). Knowing the head pose of a person provides important cues on his visual focus of attention, for example if the speaker is facing the audience.
- Audio Technologies
  - Acoustic Person Tracking The intent of the acoustic source localization task described here is to locate one or more speakers within a room. Knowing the position of a speaker is useful for beamforming, as used for far field automatic speech transcription. In the lecture scenario, the task is to locate either the lecturer or a person in the audience, while in a meeting scenario we have to locate any speaker who is talking
  - Acoustic Speaker Identification: The speaker identification task is a closed-set text independent identification task. Systems have to output an identity for each test segment. The set of reference speakers is composed of seminar speakers and participants. No overlapping speech was considered in this evaluation. Train and test segments corresponding to different speech durations were used.

- Multimodal Technologies
  - Multimodal Person Tracking: The definition of the multimodal person tracking task is quite complex, as a number of conditions and constraints arising from the fusion of modalities have to be considered. On the one hand, multimodal tracking results should be comparable to those of the visual person tracking task, where multiple persons are eventually being tracked simultaneously. On the other hand, they should be comparable to those of the acoustic person tracking task, where only one person, the active speaker, is tracked at each point in time
  - Multimodal Person Identification: In multimodal person identification, the audio (speech) and video (face) modalities are jointly used to derive the identity of the speaker.

All available information of these seminars are located in `CHIL_D7_9` directory.

In plenty of seminars, the information is supplied the same. So, next the information related to ground truth files, calibration data and formats are shown:

Methodology and protocol of audio/video/multimodal person tracking tasks There are now two possible scenarios to evaluate on.

- The first one is the Seminar scenario, as defined in previous evaluations, where the objective is to track a presenter talking to an audience in a seminar.
- The second is the Interactive Seminar scenario, in which the goal is to track all attendees of a meeting interacting with each other. The main different to the first scenario is that it is now necessary to track multiple persons in each frame, requiring different algorithms, tools, metrics, labels, etc.

In both cases, the objective is to track people as a whole, but for convenience of evaluation, a person's position is defined as the coordinates of the centre of his or her head, projected to the ground. This is the required output of a tracker and will be used in computing accuracy measures.

The multimodal evaluation is made for two conditions:

- Condition 1:
  - For the Seminar Scenario, the goal is to track the presenter for all time frames in the test sequence. Both audio and visual cues can be used to increase the precision and robustness of the tracking result.
  - For the Interactive Seminar Scenario, the goal is to track all the attendees of a small meeting and all other persons present in the room for all time frames in the test sequence. Obviously, mostly visual cues will be used here, with audio cues helping to increase tracking accuracy for the current speakers at each point in time. The results should be best comparable to those of the visual person tracking task, and the labels and metrics used should be identical.
- Condition 2:
  - For the Seminar Scenario, the objective is to track the presenter only during the time frames where he is actively speaking.

- For the Interactive Seminar Scenario, the objective is, for every time frame where speech is present, to track the person who is speaking at that point in time. These results are comparable to those of the acoustic person tracking task, and to those of the visual person tracking task, if appropriate subsets of the visual person tracking results are created, based on the speech activity detection labels and the active speaker flags included in the video labels for the multimodal task. As it is not intended to determine speech segments automatically in this task, manual labels marking the speech segments, as defined for the speech activity detection task, are used and the evaluation of the multimodal tracker performance made only on speech segments longer than 1 second.

Format of label files of audio/video person tracking tasks The labels for the visual tracking task differ according to the scenario:

1. Seminars: Required labels are the 3D positions over time of the head centroid of the presenter.
2. Interactive Seminars: Required are the 3D positions of the head centroids of all the participants of the meetings and of any other person present in the room.

The result should be 3D position labels in mm, relative to the room coordinate system. These labels should be produced for every second of video (every 15 or 25 frames, depending on the framerate). Please check the CLEAR 2006 Evaluation Plan in the `DOC` directory for more information about this.

These ground truth files for video, audio and multimodal files are in the following directories:

- AUDIO ->TECHNOLOGIES/AUDIO/ACOUSTIC_PERSON_TRACKING/REFERENCE
- VIDEO ->TECHNOLOGIES/VIDEO/VIDEO_PERSON_TRACKING/REFERENCE
- MM ->TECHNOLOGIES/MULTIMODAL/MULTIMODAL_PERSON_TRACKING/REFERENCE

Inside these directories, we will find the reference for each seminar. For the audio tasks the reference is called `seminar_active3d_label.txt`, for the video tasks `seminar_3d_label.txt` and for multimodal tasks `seminar|segment_sloc.txt` Moreover, in these reference files the times are computed as absolute time in seconds from 1.1.1970 and the coordinates are in mm.

The times in both audio and video ground truth files are absolute. However, in the video references there are more data that in the audio files. This is owing to more timestamps have been considered.

Calibration data The calibration information of each camera of site (IBM, ITC, UKA, UPC) are included under `CALIBRATION_INFORMATION` in the `DATA/<SITE>` directory.

Available results There are two `results.txt` located in `CHIL_D7_9/TECHNOLOGIES/AUDIO/ACOUSTIC_PERSON_TRACKING/RESULTS` and `CHIL_D7_9/TECHNOLOGIES/VIDEO/V` for audio and video technologies.

For multimodal tasks no results are provided.

9.1.4.2. UKA-ISL

During the seminars, video recording of the speakers and the audience from 4 fixed, calibrated cameras were made. In addition, frontal close ups of the main speaker were made with two PTZ cameras. For the audio recordings, a Countryman E6 close talking microphone was used for the main speaker, and two or three members of the audience. Far field recordings were made with NIST Mark III 64-channel microphone array, as well as four T-shaped arrays and four omnidirectional tabletop microphones.

The audio recordings of this seminars were made with:

- the CTM channel
- a 64-channels Sphere file from the corrected NIST MarkIII microphone array
- 4 T-shaped arrays of 4 channels each recording from the surrounding walls.
- 4 single-channel tabletop microphones on the central table.

Available data A total of 29 seminars were recorded, but only 12 of them are used to evaluation tasks. The rest of seminars belong to development set.

Next, the list shows the evaluation set for the evaluated technologies except for Head Pose Estimation:

- UKA_20050420_A
- UKA_20050427_B
- UKA_20050504_A
- UKA_20050504_B
- UKA_20050511
- UKA_20050525_A
- UKA_20050525_B
- UKA_20050525_C
- UKA_20050601
- UKA_20050615_A
- UKA_20050622_B
- UKA_20050622_C

Following, the development set is shown:

- UKA_20041123_A

- UKA_20041123_B
- UKA_20041123_C
- UKA_20041123_D
- UKA_20041123_E
- UKA_20041124_A
- UKA_20041124_B
- UKA_20050112
- UKA_20050126
- UKA_20050127
- UKA_20050128
- UKA_20050202
- UKA_20050209
- UKA_20050214
- UKA_20050310_A
- UKA_20050310_B
- UKA_20050314

For each evaluated seminar exists a directory with the two segments of 5 minutes recorded, named segment 1 and segment 2. Inside the corresponding segment directory, the audio and video files are located.

All these data are located into `CHIL_D7_9/DATA/UKA/AUDIO` and CHIL_D7_9/DATA/UKA/VIDEO.

The number of frames of plenty of video segments is 4502, except Segment 2 of UKA_20050525_A and Segment 2 of UKA_20050525_C with 4503 frames.

9.1.4.3. ITC-IRST

8 different cameras are used for video acquisition.

The audio recordings of this seminars were made with:

- the CTM channel
- a 64-channels Sphere file from the corrected NIST MarkIII microphone array
- 7 T-shaped arrays of 4 channels each recording from the surrounding walls.
- 4 single-channel tabletop microphones on the central table.

Available data ITC recorded 6 seminars. The two following ones were used for the evaluation tasks:

- ITC_20050503
- ITC_20050607

  Following, the development set is shown:

  - ITC_20050429

Only one segment (Segment 1) for seminar exists and the audio and video files are located in `CHIL_D7_9/DATA/ITC`.

Next, the number of frames of Segment 1 of each seminar is shown:

Seminar	Segments	Frames
ITC_20050503	1	4479
ITC_20050607	1	4487

9.1.4.4. AIT-RESIT

The sounds are recorded through the following sensors:

- 3 T-Shaped Arrays x 4 Microphones: 12 Channels in WAV format.
- 4 Table top (cardioids facing participants) microphones: 4 Channels in WAV format.
- 4 Lapel microphones: 4 Channels in WAV format.
- MARK III Array: 64 Channels in RAW format (time-stamps provided).

The video acquisition is made by 6 cameras:

- 4 Corner cameras: JPEG format.
- 1 fish-eye top camera: JPEG format.

Available data AIT-RESIT recorded 5 interactive seminars, but only 4 of them are used in the evaluation tasks:

- AIT_20051010
- AIT_20051011_B
- AIT_20051011_C
- AIT_20051011_D

Following, the development set is shown:

- AIT_20050726

There are 5 minutes per seminar recorded, only Segment 1 exists. Audio and video files are located in `CHIL_D7_9/DATA/AIT`.

Next, the number of frames of Segment 1 of each seminar is shown:

Seminar	Segments	Frames
AIT_20051010	1	8986
AIT_20051011_B	1	9003
AIT_20051011_C	1	8989
AIT_20051011_D	1	9002

9.1.4.5. UPC

The sounds are recorded by 88 different sensors:

- Hammer_01 to Hammer_12 T-shaped microphone clusters.
- Hammer_13 to Hammer_16 Omni-directional microphones on the table.
- Hammer_17 to Hammer_19 Directional microphones on the table.
- Hammer_20 to Hammer_24 CT microphones.
- MkIII_001 to MkIII_064 Mark III array microphones.

The video acquisition is done by 6 different cameras:

- 4 fixed cameras at the room corners (cam1..4).
- A zenithal fish-eye camera (Zcam5).
- An active cam (PTZ) aimed at the door (Acam8), not available for all the seminar.

Available data UPC has collected 5 interactive seminars. Each seminar consists of only a segment of 10-20 minutes presentation to a group of 3-5 attendees in a meeting room.

Only 4 seminars have been used for evaluation tasks:

- UPC_20050706
- UPC_20050720
- UPC_20050722
- UPC_20050727

Following, the development set is shown:

- UPC_20050601

The available data are located in `CHIL_D7_9/DATA/UPC` directory. The number of frames of the segment of every seminar is 7501.

9.1.4.6. IBM

The smart room sensors used for recording are:

- 5 fixed Firewire cameras (cam1-5), pixel-synchronized at 15 fps, their data saved as sequences of JPEG images, with associated time stamps. Four of these cameras (cam1-4) are located close to the corners of the room with approximately a 70o field- of-view, providing 640 x 480 pixel video frames, whereas the fifth camera (cam5) is located at the ceiling, fitted with a fish-eye lens, providing 1024 x 768 pixel frames. Calibration (both extrinsic and intrinsic) information is provided for each of these cameras.
- Two PTZ cameras (cam6-7) aiming towards the presenter area. They provide 640 x 480 pixel video frames at 30 fps, saved in the same format as above. Two additional PTZ cameras (cam8-9) are currently not being used. Cam6 is active, manually panned, zoomed and tilted so that it captures the presenterś head, whereas cam7 is fixed covering a narrow area close to the projection screen and whiteboard 1. Calibration information is provided for cam7.
- 24 synchronous channels of audio, saved as 24 bit wav files at 44 kHz. Among these, 19 channels provide far-field signals and five are close-talking ones. The former correspond to four T-shaped microphone arrays, each having four microphones, located on the two long walls of the IBM room. The remaining three channels are from table-top microphones located on the conference table. Finally, from the five close-talking microphones, four are wired, worn by participants seated around the table, and one is wireless, worn by the seminar presenter. Timing information is also provided marking the beginning and end of the recording.
- Two NIST MkIII microphone arrays, each generating 64 channels of audio, captured at 44 kHz and 24 bits of resolution. Channel 4 of each is extracted in raw format. Note that timing information is provided containing the beginning and ending timestamps of each array recording.

Available data Five seminars have been recorded in this room. The seminars have duration of approximately 30 minutes each, with only a small number of audience members (typically 3-5). Four of the seminars have been used to evaluation tasks, with only Segment 1:

- IBM_20050819
- IBM_20050822
- IBM_20050823
- IBM_20050830

Following, the development set is shown:

- IBM_20050824

All these data are located in `CHIL_D7_9/DATA/IBM` directory. The number of frames of all the seminars is 4501.

9.1.5. 2007 CHIL Evaluation Campaign

9.1.5.1. Introduction

This document specifies in details the Evaluation Plan for the technological areas of CHIL and for their technological components, which are to be evaluated during the CLEAR07 Evaluation Workshop. The Evaluation tests will be run on the data provided by IBM, Istituto Trentino di Cultura (ITC- resit), Research and Education Society in Information Technology (RESIT-AIT), Universitat Politécnica de Catalunya (UPC) and University of Karlsruhe (UKA). These data mainly consist of audio and video recordings of seminars given at each site.

During the CLEAR07 evaluation campaigns, 9 technologies are evaluated:

- Vision technologies
  - 2D Face Detection / Tracking.
  - 3D Person Tracking.
  - Visual Person Identification.
  - Head Pose Estimation.
- Audio technologies.
  - Acoustic Person Tracking.
  - Acoustic Person Identification.
  - Acoustic Event Detection.
- Multimodal technologies.
  - Multimodal Person Tracking.
  - Multimodal Person Identification.

In this campaign seminar were recorded at 5 sites:

- UKA
- ITC-IRST
- UPC
- RESIT-AIT
- IBM

In these sites 5 seminars were recorded, and there are two segments of 5 minutes (A and B) per seminar.

All available data of theses seminars are located in
`CHIL 2007/CHIL_D7_14/CHIL_D7_14/DATA/TEST/SEMINARS`

Moreover, empty room (background) sequences have also been recorded to facilitate tracking algorithms, and are located in
`DATA/TEST/SEMINARS/<Seminar>_<Segment>/info/background`

Video and microphones quality standard

- Video quality standard. The minimum frame rate was set to 15 frames per seconds. The data were saved as sequences of JPEG images in a fixed name standard: seq xxxxx.jpg with xxxxx the number of the frame. The maximum desynchronization between the five cameras was fixed to 3 frames at 15 fps.
- Microphone array quality standard. Each site is equipped with at least on fully functional Mark III microphone array version 2. Itś generating 64 channels of audio, captured at 44 KHz and 24 bits of resolution. For each recording, the channel 4 was extracted. A specific file called timestamps.ini was created to store the time stamp of an eventual packet loss. The maximum number of packet loss during one recording was fixed to 200 or 220ms. If more occurred, the recording had to be remade.

Format of label files of audio/video person tracking tasks

1. Description of label file formats for vision technologies.
   In general, it has been agreed to produce manual annotations, labels, only every second, i.e. every 15th, 25th or 30th video frame, depending on the frame rate.

   - Generic 2D labels. For the 5 seminars of the development set, the 20 first minutes were labelled For each of the 20 seminars of the evaluation set, 2 fiveÂminute segments were selected by UKA. In total, 40 5Âminute video segments (200 minutes) were labelled for the evaluation set. Video labelling was done for the 5 CHILÂroom cameras: the 4 corners cameras of the CHILÂ rooms (cam1 to cam4), and the ceiling camera (cam5). The following labels were produced every 1 second, for each camera and each person attending the seminars:
     - Head Centroid,
     - Face Bounding Box.
     - Nose Bridge,
     - Right Eye,
     - Left Eye.

     Only the Head Centroid label was produced for the ceiling camera (cam5). In each seminar, all attendees were labelled.
   - 3D labels files. For the evaluation of some tasks, namely the 3D audio, visual and AV person tracking tasks, the visual and AV person identification tasks, and the head pose estimation task, the 3D positions of persons in the room must be known. Therefore, a separate label file, describing the 3D positions of the seminar participantsḣead centroids in the room coordinate frame will be generated from the 2D label files using calibration and room layout information provided by the recording sites.

   The ground truth files are of each tecnhology are located in `DATA/TEST/SEMINARS/video_labels`. Inside this directory there are the label files of audio, video and multimodal technologies, and they are named:

   - AUDIO: `<Seminar>_<Segment>_Acoustic3d_label.txt`
   - VIDEO: `<Seminar>_<Segment>_3d_label.txt` and `<Seminar>_<Segment>_camX_label.txt`.
   - MM: `<Seminar>_<Segment>_Multimodal3d_label.txt`

In all these reference files the times are computed as absolute time in seconds from 1.1.1970 and the coordinates are in mm. In a concrete segment, the length of these three reference files may be different, and this is owing to more timestamps have been considered.

Calibration data The available calibration information are located in `DATA/TEST/SEMINARS/<Seminar>_<Segment>/info/calibration`.

9.1.5.2. UKA-ISL

9.1.5.3. Available data

A total of 5 seminars were recorded at Karlsruhe:

- UKA_20060726
- UKA_20060912
- UKA_20061116
- UKA_20061120
- UKA_20061207

The seminar UKA_20060726 belongs to development set and the rest to evaluation/test set. Therefore only 4 seminar have been used in evaluation tasks.

Next, the number of frames of two segments of each seminar is shown:

Seminar	Segments	Frames
UKA_20060912	A	4502
	B	4623
UKA_20061116	A	4517
	B	4472
UKA_20061120	A	4636
	B	4547
UKA_20061207	A	4577
	B	4727

9.1.5.4. ITC_IRST

Available data A total of 5 seminars were recorded at ITC:

- ITC_20060714
- ITC_20060922A
- ITC_20060922B
- ITC_20060927
- ITC_20060928

The seminar ITC_20060714 belongs to development set and the rest to evaluation/test set.

Next, the number of frames of two segments of each seminar is shown:

Seminar	Segments	Frames
ITC_20060922A	A	4585
	B	4704
ITC_20060922B	A	4660
	B	4540
ITC_20060927	A	4555
	B	4570
ITC_20060928	A	4884
	B	4585

9.1.5.5. UPC

Available data UPC has collected 5 interactive seminars:

- UPC_20060613
- UPC_20060620
- UPC_20060713
- UPC_20060720
- UPC_20060720B

The seminar UPC_20060613 belongs to development set and the rest to evaluation/test set.

Next, the number of frames of two segments of each seminar is shown:

Seminar	Segments	Frames
UPC_20060620	A	7501
	B	8001
UPC_20060713	A	7601
	B	7626
UPC_20060720	A	7726
	B	7851
UPC_20060720B	A	7551
	B	8176

9.1.5.6. RESIT-AIT

Available data RESITÂAIT has recorded 5 interactive seminars:

- AIT_20060728
- AIT_20061020

- AIT_20061020B
- AIT_20061020C
- AIT_20061020D

The seminar AIT_20060728 belongs to development set and the rest to evaluation/test set.

Next, the number of frames of two segments of each seminar is shown:

Seminar	Segments	Frames
AIT_20061020	A	9303
	B	9363
AIT_20061020B	A	9003
	B	9122
AIT_20061020C	A	9033
	B	9422
AIT_20061020D	A	9153
	B	9123

9.1.5.7. IBM

In this seminar two additional panÂtiltÂzoom (PTZ) cameras have been activated for data collection, bringing the total number of recording cameras to nine (five fixed, four PTZs). Each pair of PTZ cameras requires a dedicated computer to allow recording of 640x480Âpixel video at 30 Hz.

An effort has been made that the recordings demonstrate a high degree of interactivity (questions, answers, discussion), person movement (typically participants entering or exiting the room), and a significant number of acoustic events (cellÂphone rings, door knocks, typing noise, etc.).

Available data IBM has recorded 5 interactive seminar:

- IBM_20060720
- IBM_20060810
- IBM_20060811
- IBM_20060814
- IBM_20060815

The seminar IBM_20060720 belongs to development set and the rest to evaluation/test set.

Next, the number of frames of two segments of each seminar is shown:

Seminar	Segments	Frames
IBM_20060810	A	4502
	B	4502
IBM_20060811	A	4547
	B	4501
IBM_20060814	A	4651
	B	4621
IBM_20060815	A	4532
	B	4577

Bibliografía

Bibliografía

[1] S. of the art overview: Localization and tracking of multiple interlocutors with multiple sensors, "Augmented multi-party interacion (ami) project." Technical report, Tech. Rep., 2007.

[2] M. L. Seltzer, "Microphone array processing for robust sppech recognition," *Carnegie Mellon University Pittsburgh*, 2003.

[3] W. Herbordt, *Sound capture for human/machine interfaces - Practical aspects of microphone array signal processing*. Springer, Heidelberg, Mar. 2005.

[4] D. Gelbart and N. Morgan, "Double the trouble: Handling noise and reverberation in far-field automatic speech recognition." *Inernation Conference on Spoken Language Processing (ISCLP)*, 2002.

[5] S. Kochkin and T. Wickstrom, "Headsets, far field and handheld microphones: Their impact on continuous speech recognition," *Technical report, EMKAY, a division of Knowles Electronics*, 2002.

[6] A. Vinciarelli, M. Pantic, H. Bourlard, and A. Pentland, "Social signal processing: state of the art and future perspectives of an emerging domain," *Proceeding of the 16th ACM international conference on Multimedia*, pp. 1061–1070, 2008.

[7] E. M. Herraiz, "Diseño, implementación y evaluación de técnicas de localización y de mejora de la señal de habla en entornos acústicos reverberantes: aplicación a sistemas de reconocimiento automático de habla." Master's thesis, ETSI Telecomunicación, Universidad Politécnica de Madrid, Spain, 2005.

[8] C. Castro García, "Speaker localization techniques in reverberant acoustic environments," Master's thesis, Royal Institute of Technology (KTH), Stockholm, 2007.

[9] M. C. Aguilar, "Comparativa teórica y empírica de métodos de estimación de la posición dfe múltiples objetos," Master's thesis, Escuela Politécnica de Alcalá de Henares, Oct. 2007.

[10] M. Marrón, "Seguimiento de múltiples objetos en entornos interiores muy poblados basado en la combinación de métodos probabilísticos y determinísticos," Ph.D. dissertation, Departamento de electrónica, Universidad de Alcalá, 2008.

[11] D7.4 evaluation packages for the first chil evaluation campaign.

[12] E. Aarts, H. Harwig, and M. Schuurmans, "Ambient intelligence," in *The Invisible Future*. ed., McGraw Hill, 2001, pp. 235–250.

[13] K. Okuma, A. Taleghani, N. De Freitas, J. Little, and D. Lowe, "A boosted particle filter: multi-target detection and tracking," *Proceedings of the Eight European Conference on Computer Vision (ECCV04), Lecture Notes in Computer Science*, vol. 3021, pp. 28–39, May 2004.

[14] O. Lanz, "Aproximate bayesian multibody tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 1436–1449, Sept. 2006.

[15] G. Lathoud and M. Magimai-Doss, "A sector-based, frequency-domain approach to detection and localization of multiple speakers," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 265–268, Mar. 2005.

[16] D. Gatica-Perez, G. Lathoud, J.-M. Odobez, and I. McCowan, "Audio-visual probabilistic tracking of multiple speakers in meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 601–616, Feb. 2007.

[17] Amida, *Augmented Multi-party Interaction with Distance Access. Localization and Tracking of Multiple Interlocutors with Multiple Sensors.* January, 2007.

[18] M. Brandstein and D. Ward, *Microphone Arrays.* Springer, 2001.

[19] H. Buchner, R. Aichner, and W. Kellerman, "Simultaneous localization of multiple sound sources using blind adaptive mimo filtering," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. 18–23, Mar. 2005.

[20] M. Matsumoto, "A study on miniaturizing microphone array utilizing aggregated microphones," *School of Science and Engineering Waseda University*, Mar. 2006.

[21] M. Omologo, L. Arman, M. Matassoni, and P. Sveizer, "Use of a csp-based voice activity detector for distant-talking asr," *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech).*, pp. 501–504, 2003.

[22] G. Lathoud, I. McCowan, and J. Odobez, "Unsupervised location-based segmentation of multi-party speech," *Proceedings of NIST-ICASSP Meeting Recognition Workshop*, Apr. 2004.

[23] I. A. McCowan, "Robust speech recognition using microphone arrays," Ph.D. dissertation, Queensland University of Technology, Australia, 2001.

[24] M. Omologo and P. Svaizer, "Acoustic event localization using a crosspower-spectrum phase based technique," *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 273–276, Apr. 1994.

[25] K. Varma, "Time-delay-estimate based direction-of-arrival estimation for speech in reverberant environments," Ph.D. dissertation, Virginia Polytechnic Institute, 2002.

[26] B. De Moor, "The singular value decomposition and long and short spaces of noisy matrices," *IEEE Transactions on Signals Processing*, vol. 41, pp. 2826–2838, Sept. 1993.

[27] B. Mungamuru and P. Aarabi, "Enhanced sound localization," *IEEE Transactions Systems, Man, and Cybernetics*, vol. 34, pp. 1526–1540, June 2004.

[28] A. Brutti, M. Omologo, and P. Svaizer, "Oriented global coherence field for the estimation of the head orientation in smart rooms equipped with distributed microphone arrays," *INTERSPEECH 2005*, pp. 2337–2340, Sept. 2005.

[29] F. Abad, "A multi-microphone approach to speech processing in a smart-room environment," Ph.D. dissertation, Universidad Politécnica de Cataluña, Feb. 2007.

[30] R. M. Stern, "Using computational models of binaural hearing to improve automatic speech recognition: Promise, progress and problems," *AFORS workshop on Computational Audition*, 2002.

[31] D. Li and S. Levinson, "A bayes-rule based hierarchical system for binaural sound source localization," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, Apr. 2003.

[32] F. Keyrouz and K. Diepold, "An enhanced binaural 3d sound localization algorithm," *IEEE International Symposium on Signal Processing and Information Technology*, pp. 662—-665, Aug. 2006.

[33] X. Chen, X. Hao, R. Wu, X. Wu, and S. Zhao, "Binaural sound source localization based on steered beamformer with spherical scatterer," *30th International Conference: Intelligent Audio Environments*, 2007.

[34] S. Benton and A. Spanias, "Effects of reverberation on sound source localization using binaural spectral cues," *Proceeding (412) Modelling, Identification, and Control*, 2004.

[35] J. H. Dibiase, "A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays," Ph.D. dissertation, Brown University, 2000.

[36] B. D. V. Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE Signal Processing Magazine*, vol. 5, pp. 4–24, Apr. 1988.

[37] F. J. R. Fernández, "Design, implementation and evaluation of techniques for speech signal improvement in reverberant environments: application in automatic speech recognition systems," Ph.D. dissertation, Technical University of Madrid, 2007.

[38] L. J. Ziomek, "Fundamental of acoustic field theory and space-time signal processing," *CRC Press*, 1995.

[39] D. Zotkin, R. Duraiswami, and L. S. Davis, "Active speech source localization by a dual coarse-to-fine search," *IEEE Internacional Conference on Acoustic, Speech and Singal Processing*, vol. 5, pp. 3309–3312, May 2001.

[40] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, pp. 5–28, June 1998.

[41] T. Zhao and R.Ñevatia, "Tracking multiple humans in crowded environment," *Proceedings of the Third IEEE Conference on Computer Vision and Pattern Recognition (CVPR04)*, vol. 2, pp. 406–413, June 2004.

[42] W.Ñg, J. Li, and J. Godsill, "Online multisensor-multitarget detection and tracking," *Proceedings of the 2006 IEEE Aerospace Conference*, Mar. 2006.

[43] C. Hue, P. Le Cadre, and P. Pérez, "A particle filter to track multiple objects," *IEEE Workshop on Multi-Object Tracking*, pp. 61–68, July 2001.

[44] S. Thrun, "Probabilistic algorithms in robotics," *Artificial Intelligent Magazine*, vol. 21, pp. 93–109, Apr. 2000.

[45] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, Feb. 2002.

[46] C. Rasmussen and G. Hager, "Probabilistic data association methods for tracking complex visual objetcs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 560–576, June 2001.

[47] D. Schulz, W. Burgard, and D. Fox, "Tracking multiple moving objects with a mobile robot," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, pp. 371–377, Dec. 2001.

[48] O. Lanz and R. Brunelli, "Dynamic head location and pose from video," *IEEE International Conference on Multisensor Fusion and Integration from Intelligent Systems*, pp. 47–52, Sept. 2006.

[49] S. Ba and J. Odobez, "A probabilistic framework for joint head tracking and pose estimation," *Proceedings of the 17th Int.Conf. on Pattern Recognition*, vol. 4, pp. 264–267, Aug. 2004.

[50] S. Ba and J. M. Odobez, "A rao-blackwellized mixed state particle fitler for head pose tracking," *ICMI Workshop on Multimodal Multiparty Meetings*, pp. 9–16, 2005.

[51] S. Ba and J. Odobez, "Multi-party focus of attention recognition in meetings from head pose and multimodal contextual cues," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, vol. 4, pp. 2221–2224, Mar. 2008.

[52] J. Odobez and S. Ba, "Cognitive and unsupervised map adaptation approach to the recognition of the focus of attention from head pose," *IEEE International Conference on Multimedia and Expo 2007*, pp. 1379–1382, July 2007.

[53] K. Smith, S. Ba, J. Odobez, and D. Gatica-Perez, "Tracking the visual focus of attention for a varing number of wandering people," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1212–1229, July 2008.

[54] V. E. Gómez, "Sistema de detección de obstáculos y robots mediante múltiples cámaras en espacios inteligentes," Master's thesis, Escuela Politécnica de Alacalá de Henares, Dec. 2008.

[55] R. Hartley and A. Zisserman, *Multiple View Geometry in computer vision.* Ed. Cambridge, 2003.

[56] M. Marrón, M. Sotelo, J. García, D. Fernández, and D. Pizarro, "Xpfcp: An extended particle filter for tracking multiple and dynamic objects in complex environments," *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS05)*, p. 234..239, Aug. 2005.

[57] A. Almeida, J. Almeida, and R. Araújo, "Real-time of moving objects using particle filter," *Institute for Systems and Robotics. Department of Electrical and Computer Engineering, University of Coimbra, Coimbra, Portugal*, 2005.

[58] K. Smith, D. Gatica-Perez, and J. Odobez, "Using particles to track varying numbers of interacting people," *Proc. of the Fourth IEEE Conference on Computer Vision and Pattern Recognition (CVPR05)*, pp. 962–969, June 2005.

[59] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEEE Proceedings Part F*, vol. 140, no. 2, pp. 107–113, Apr. 1993.

[60] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *Proc. of the Seventh IEEE Inernational Conference on Computer Vision(ICCV99)*, vol. 1, pp. 572–578, Sept. 1999.

[61] M. Marrón, M. Sotelo, J. García, and J. Broddfelt, "Comparing improved versions of k-means and substractive clustering in a tracking applications," *Proc. of the Eleventh International Workshop on Computer Aided Systems Theory, Extended Abstracts (EUROCAST07)*, pp. 252–255, Feb. 2007.

[62] J. C. Jiménez, "Comparativa teórica y empírica de métodos de clasificación aplicados a medidas tridimensionales de visión," Master's thesis, Universidad de Alcalá, July 2007.

[63] G. Potamianos, C.Ñeti, G. Gravier, A. Garg, and A. W. Senio, "Recent advances in the automatic recognition of audiovisual speech," *Proc. of the IEEE*, vol. 91, no. 9, pp. 1306–1326, Sept. 2003.

[64] J. G. Pérez, A. Frangi, E. Lleida, and K. Lukas, "Lip reading for robust speech recognition on embbeded devices," *International Conference on Acoustic, Speech and Signal Processing, ICASSP-05*, Mar. 2005.

[65] G. Potamianos, C.Ñeti, J. Luettin, and I. Matthews, "Audio-visual automatic speech recognition: An overview," *Issues in Visual and Audio-Visual Speech Processing*, 2004.

[66] P. Aarabi and S. Zaky, "Robust sound localization using multi-source audiovisual information fusion," *Inf. Fusion*, vol. 3, no. 2, pp. 209—-223, Sept. 2001.

[67] R. Cutler and L. Davis, "Look who is talking: Speaker detection using video and audio correlation," *Proc. IEEE Int. Conf. Multimedia (ICME)*, pp. 1589—-1592, July 2000.

[68] J. Hershey and J. Movellan, "Audio vision: Using audio–visual synchrony to locate sounds," *Proc. Neural Inf. Process. Syst. (NIPS)*, pp. 813—-819, Nov. 1999.

[69] J. Vermaak, M. Gagnet, A. Blake, and P. Perez, "Sequential monte carlo fusion of sound and vision for speaker tracking," *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, pp. 741—-746, July 2001.

[70] *CHIL, Deliverable D7.7 Specification of Evaluation Packages: Report about Data, Tasks and Metrics*, 2007.

[71] M. Siracusa, L.-P. Morency, K. Wilson, J. Fisher, and T. Darrell, "A multi-modal approach for determining speaker location and focus," *Proc. Int. Conf. Multimodal Interfaces (ICMI)*, pp. 77—-80, 2004.

[72] Y. Chen and Y. Rui, "Real-time speaker tracking using particle filter sensor fusion," *Proc. IEEE*, vol. 92, no. 3, pp. 485—-494, Mar. 2004.

[73] N. Checka, K. Wilson, M. Siracusa, and T. Darrell, "Multiple person and speaker activity tracking with a particle filter," *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.(ICASSP)*, pp. 881—-884, May 2004.

[74] *Deliverable D 7.7 Specification of Evaluation Packages: Report about Data, Tasks and Metrics*, 2006.

[75] R. Stiefelhagen and J. Garofolo, *Multimodal Technologies for Perception of Humans.* Springer, 2006.

[76] R. Stiefelhagen, R. Bowers, and J. Fiscus, *Multimodal Technologies for Perception of Humans*, ser. International Evaluation Workshops CLEAR 2007 and RT 2007. Baltimore, MD, USA: Springer, May 2007.

[77] M. Wax and T. Dailath, "Optimum localization of multiple sources by passive arrays," *IEEE Transaction on Acoustics, Speech, and Signal Processing*, vol. 31, pp. 1210–1217, Oct. 1983.

[78] D. Zotkin and R. Duraiswami, "Accelerated speech source localization via a hierarchical search of steered response power," *IEEE Transaction on Acoustics, Speech, and Signal Processing*, vol. 12, pp. 499–508, Sept. 2004.

[79] R. Kalman, *A new approach to linear filtering and prediction problems.* Transactions of the ASME-Journal of Basic Engineering, Mar. 1960, vol. 82.

[80] G. Welch and G. Bishop, "An introduction to the kalman filter," *TR 95-041, Department of Computer Science University of North Carolina at Chapel Hill*, July 2006.

[81] C.Ñadeu, C. Segura, D. Abad, and J. Hernando, "Multimodal person tracking in a smart-room environment," *IV Conference on Speech Technology*, pp. 271–276, Nov. 2006.

[82] E. Lehmann, "Particle filtering methods for acoustic source localization and tracking," Ph.D. dissertation, Australian National University, July 2004.