Video conference smart room: an information fusion system based on distributed sensors

Khairunizam Wan, Atsushi Todo, Hideyuki Sawada, Olivier Passalacqua, Eric Benoit, Marc-Philippe Huget, Patrice Moreaux

Video conference smart room: an information fusion system based on distributed sensors

Khairunizam WAN
Atsushi TODO
Hideyuki SAWADA
Faculty of Engineering,
Kagawa University, JAPAN
Email: sawada@eng.kagawa-u.ac.jp,
(TODO) S07g525@stmail.eng.kagawa-u.ac.jp,
(WAN) s06d507@stmail.eng.kagawa-u.ac.jp

Olivier PASSALACQUA
Eric BENOIT
Marc-Philippe HUGET
Patrice MOREAUX
Laboratoire d'Informatique, Systèmes,
Traitement de l'Information et de la Connaissance
Université de Savoie, B.P. 80439,
74944 Annecy le Vieux Cedex, FRANCE
Email:{olivier.passalacqua, eric.benoit,
marc-philippe.huget, patrice.moreaux}@univ-savoie.fr

***Abstract*— The needs for cross domain technologies called mecatronics have increased in the recent years and examples of mechanical tools directed by computers are now widely available. This paper gives an example of information fusion in the context of a video conference room and exposes two axes of the research. First it shows how to fuse information provided by several sources to locate a speaker. To do this, the system fuses data produced by video cameras and their associated image processing algorithm, with information resulting from signal processing algorithms applied on several microphones. Second, this article describes the distributed information fusion system (DIFS) used and the algorithm which decides where the speaker is located in order to allow focus on him. The whole application is managed by a new control system specifically developed for DIFSs. Some key points of the theoretical model on which the control is based are also given. This project has been realized thanks to an international cooperation between the Kagawa University in Japan and the University of Savoie in France.**

KEYWORDS distributed system, image and signal processing, information fusion.

INTRODUCTION

The present work deals with identification and localization of people in the context of a video conference room. Several information sources with different properties and several computation resources are used together for this application. A typical disposition of our video conference room is shown in Figure 1. Due to there specific properties, a sound sensing system and a video capture system are involved. Each one provides partial information about people in the room and information fusion (or aggregation) between the two systems allows us to control a video camera tracking the identified speaker (the speaker's name is written on the picture) for remote participants.

The sound system first locates the direction of the sound source and computes the sound characteristics of the user who is speaking in the group of people. These characteristics allow identification of the speaker using a comparison algorithm.

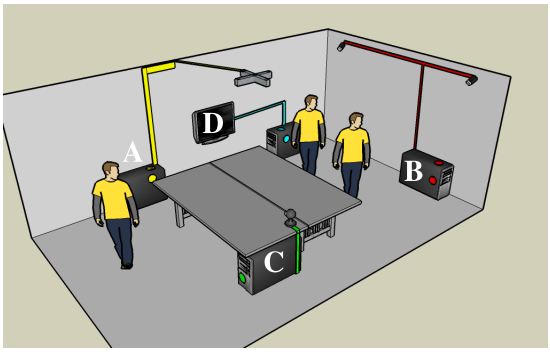

Fig. 1. Video conference room experiment overview. The video system (B) and the sound system (A) provide the locations of the users and the identity of the speaker. The computer C fuses the information and provides speaker's location. Finally a camera focuses on the speaker and sends the video to the computer D.

At the same time an optical motion capture system gives the precise location of the users by tracking reflective markers attached to their heads. Then dedicated functions group the markers by couple and compute the centre of gravity of each head.

Information produced by each source is then incomplete. The video based 3D acquisition system gives the precise location of the users but it is not able to identify who is speaking. Following the work of [FKI+04], the sound system is able to provide the direction of the sound source and can identify the speaker"s voice signature, but it cannot distinguish between a user and a loudspeaker. Moreover, since the various functions of the system run on different computers, we are indeed faced to the design and the implementation of a *Distributed - Information Fusion System* (D-IFS).

Information fusion is the process of computing "higher level" or/and a "better" information from several orginal information sources. Evaluation of the information level refers to the scale from the hardware level to the Artificial Intelligence level. For what concern comparison of information quality, it should

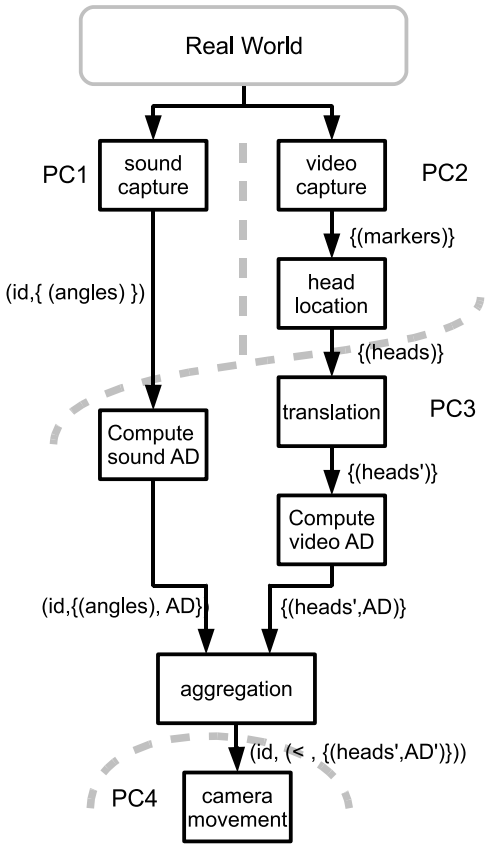

Fig. 2. Information Fusion System for the Video Conference Room

be defined through adapted and computable criteria like precision, certainty, etc. Designing an effective D-IFS first requires to describe the fusion process and then to solve the deployment problem, i.e. how and where to run the components of the system in a distributed environment. In this paper, we anlayze these different steps and we present our solution based on OSGi [Kno], [OSG] frameworks running on each computer. This software architecture allows us to control the execution of the fusion process (fusion functions and communications among them).

The paper is organized as follows. Section I addresses the description of the information fusion system used in the project. In Section II, we describe the algorithms for image processing and Section III details the sound signal processing before fusing information. Section IV explains the proposed fusion method used in the experiments presented in Section V.We summarize our results and present futur works in the conclusion.

I. Information Fusion System

As in our previous work [BHMP07] on IFS, we describe an IFS as a discrete data-flow (directed) graph. Each node of the graph is a *fusion function*. Such a function consumes information source(s) on *data input port(s)*, produces information on *data output port(s)* depending on parameters defined through *parameter input port(s)*. The IFS is then fully defined by the set of its fusion functions $F = \{f_i \mid i \in I_F\}$ and links between output ports, input ports and parameter input ports of the (f_i).

Figure 2 gives the compact description of the IFS we have developed for the video conference room (PCi and gray dashed lines will be explained in section V). The two capture functions are connected to the sensors. The output of the *sound capture* is $(id, \{(\alpha_j, \theta_j) \mid j \in J_d\})$, where id is the template identified by the sound system and (α_j, θ_j) are the sound direction angles for directions set J_d. To avoid wrong selection as in the case of an echo, the sound system provides the list of all the possible sound directions detected.

The output of the *video capture* is a set of marker 3D coordinates $C = \{c_i \mid i \in I_c\}$ where I_c is the set of markers. The *head location* function groups the markers by couple and computes the centre of gravity of each user's head as explained in Section II. The *translation* function translates the location of the users from video capture referential to global 3D coordinates in order to allow aggregation. It provides a set of user coordinates $G = \{g_i \mid i \in I_g\}$ where I_g is the set of users.

An accurateness degree denoted by AD is assigned to the information provided by each source. The AD function should provide a real number between 0 (low quality) and 1 (high quality). Such functions are frequently used in IFS to tune aggregation algorithms. They may be based on probabilistic or possibilistic or fuzzy approaches. In our context, we define AD as follows in a deterministic way:

$$\text{sound system}: \ \forall(\alpha_j, \theta_j), AD_j = \sin(\alpha_j) * \sin(\theta_j).$$
$$\text{video system}: \ \forall g_i, \ AD_i = \frac{\min_{j \neq i}\{d(g_i, g_j)\}}{\max_{j \in I}\{d(g_i, g_j\}}$$

where d is the Euclidian distance. This definition implies that AD equals 1 when the speaker is in front of the sound system, and for the tuples of coordinates, AD is near 0 when the users are grouped and near 1 when the users are far from each other.

The final step, *aggregation* of the fusion process, detailed in Section IV, aggregates locations of the users and direction of the speaker.

II. Image processing

In our project, two reflective markers are fixed to each user's head. The system imports from the motion capture system a set of frames made of the marker locations and then groups the markers by couple in order to compute the Centre Of Gravity (COG) of each head.

To measure the 3D location of a speaker, an optical motion capture system connected to a capture software [KH07] tracks in real time markers through two high-speed cameras with an image resolution of 640 x 480 pixels and the ability of capturing 120 frames per second.

Based on the 3D position of each marker, algorithm 1 computes the COG of each user to determine

Algorithm 1: Grouping algorithm

Data : $C = \{c_i \mid i \in I_c\}$ the set of marker coordinates.
Result: $G = \{g_i \mid i \in I_g\}$ the set of COG coordinates.
foreach $(i,j) \in I_c^2$ **do**
  $\mathcal{D}m_{i,j} = d(c_i,c_j)$;
end
foreach $k \in I_g, (i,j) \in I_c^2, j \neq i$ **do**
  $\mathcal{D}m_{i,min} = min\{\mathcal{D}m_{i,k}\}_{\forall k \neq i}$;
  if *free*(c_j) *and* $\mathcal{D}m_{i,j} = \mathcal{D}m_{i,min}$ **then**
    lock(c_j);
    $g_k = \frac{c_j + c_i}{2}$
  end
end
return(G);

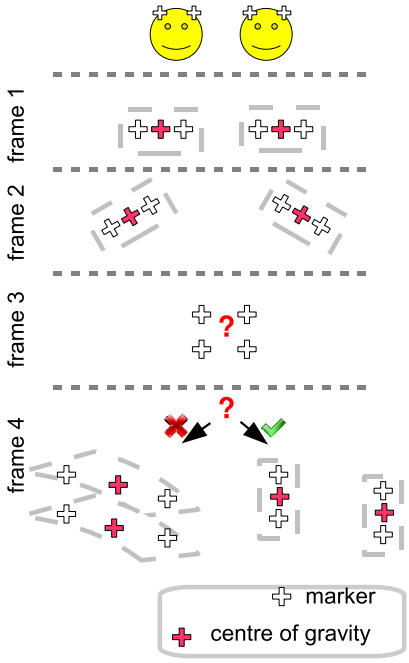

Fig. 3. Borderline cases of the grouping algorithm. The limit case of frame 3 should not append according to where the markers are fixed. Even after such case the system will provide the right marker couples as in frame 4

where each user is located in the room. This algorithm groups the markers according to the minimal distances among each couple of markers to avoid wrong position result. The sub-algorithm *free* checks if a marker is not already assigned to prevent the affectation of a marker to two users. As shown in Figure 3 the algorithm has to deal with two borderline cases. The first one (shown if frame 3) occurs when the distance between the markers of one user is equal to the distance between this user and is neighbour. This case is resolved by disposing the markers on a user's head such that their distance is negligible comparing to the distance between this user and his nearest neighbour. In the second case (shown in

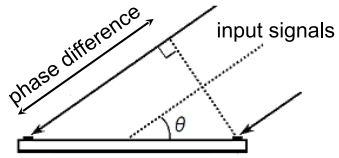

Fig. 4. The centre microphone is used to compute the sound characteristic parameters. The other four microphones are used for the sound source localisation

frame 4), the algorithm should not group the markers as shown in left side of the figure. To prevent such groups the algorithm always assign two markers in the same couple according to the smallest distance.

III. Sound signal processing

The system shown in Figure 4 has been developed by Sawada laboratory to track a 3D location of a sound source and also to identify a speaker. This identification is based on sound characteristics and the sound source location is estimated by using the phase difference and sound pressure difference among the five microphones of the system [TKTH04].

Mel-cepstrum coefficients are used for the sound identification since they present the sound characteristics. In this system, sound parameters such as spectral features, time differences, amplitudes and fundamental frequencies are extracted from the sound signals inputted from the five microphones, and used as sound feature boxes in the system.

To prevent mistakes due to the selection of only one position between several ones such as in the case of an echo, the microphone node provides a set of sound directions (α, θ). The accurateness degree is linked to the speaker direction from the sensor. The more in front of the system the speaker is, the righter his direction is.

IV. Aggregation method

The aggregation step is the final one of the processing chain. The fusion stage takes place after computation of all coordinates in the same referential and after attribution of an accurateness degree to each proposition produced by the sensors. The result of the aggregation is a sequence of user's positions, ordered with decreasing speaker indicator values (AD') which are also given. In the present fusion function, we have chosen to select a subset of the coordinates produced by the video system because these values are accurate regarding to the value of the sound direction. In case both sources provide unaccurate values, the algorithm should be modified in order to choose a computed intermediate position and not an effective one.

Input data of the fusion function are the following:

$$D = \{(\alpha_j, \theta_j, AD_j) \mid j \in J_d\} \quad \text{(sound system)}$$
$$G = \{(x_g, y_g, z_g, AD_g) \mid g \in I_g\} \quad \text{(video system)}$$

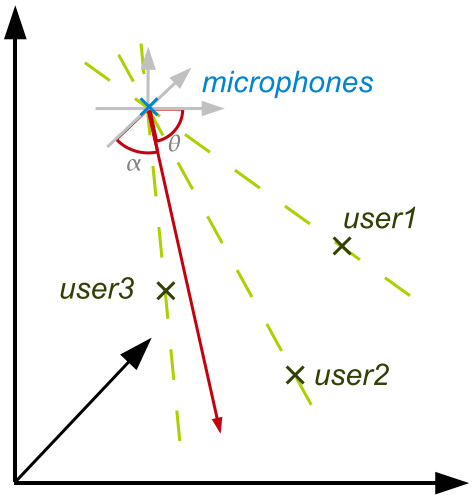

Fig. 5. All information in the same referential. The system translates the positions provided by the video system into a global referential to compare the direction of the speaker with the users ones

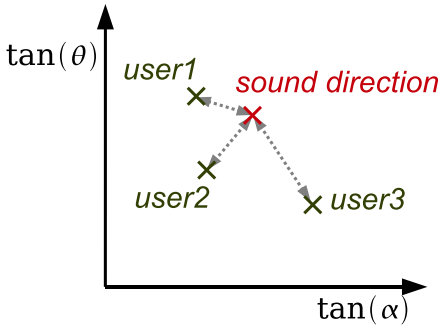

Fig. 6. Tangent values comparison: the sound source direction (red) (it is possible to have several ones); the effective positions of the users (green)

The two angles produced by the sound system represent the direction of the sound source as shown in Figure 5. The coordinates received by the fusion function are those of the COG of each user's head - a translation to the global referential having been already applied. Note that the aggregation process is required only to know the speaker's position, not his identity (given by the microphone system). However, another source of identification such as motion recognition could also be involved to increase the quality of this functionality.

The aggregation function needs parameters below to fuse input information:

(x_M, y_M, z_M) (the microphone system position)
$D_p, A_p \in \mathbb{N}$
$n_r \in \mathbb{N}$

The microphone system's position, expressed in the global referential, may be updated/modified due to physical move of the microphone. The two parameters D_p and A_p modify the impact of the effective distance on the quality indicator in the ordering algorithm. The number n_r of results produced may be changed. This could be useful when an error is detected after having moved the video camera - i.e. nobody in front of the camera. By increasing the number of solutions, the system allows the camera to move to the next possible position. Modifications/updates of these parameters are taken into account in real-time by our IFS implementation.

The fusion function first computes the equations of the straight lines passing through the microphone system and through each user as shown in Figure 5. 2D space projections of these straight lines are then obtened as $proj(\alpha, \theta) = (\tan(\alpha), \tan(\theta))$.

The fusion function then computes the *selection distances* SD between the speaker (S) and each position ((x_g, y_g, z_g)) and sorts them in the increasing order. The selection distance $SD(g)$ between (α_S, θ_S) and $(\alpha_g, \theta_g))$ takes into account the effective distance $\Delta_{S,g}$ between the two directions and the accurateness degrees AD:

$$SD(g) = (\Delta_{S,g})^{D_p} * (AD_S * AD_g)^{I_p}.$$

The final selection of the fusion function is the SD-ordered sequence C_s of the n_r coordinates with highest selection distance. The fusion function also produces the accurateness degree $AD'_g = AD_S * AD_g$ for each element of C_s. Hence, the result provided by the aggregation function is the ordered set $\{(c_g, AD'_g) \mid c_g \in C_s\}$ representing the possible speaker's positions. This result feeds a control system moving a video camera to focus on the speaker.

V. Experiments and Implementation

We conducted three experiments to validate our video conference smart room IFS and its software implementation. Physical infrastructure of the IFS comprises four personal computers (PC1 to PC4 on Figure 2). The IFS functions are implemented as indicated by grey dashed lines on Figure 2. Distribution of these functions among computers are mainly constrained by sensor dedicated software and the amount of data exchanged between the sensors and their capture functions. Moreover, since this work is also a case study for our distributed IFS implementation, we run conversion and aggregation functions on a dedicated computer (PC3): the sound system is controled by PC1. The video camera is connected to PC2, a computer dedicated to video conferencing software. Finally, PC4 drives the camera moves. These four computers are linked together through a TCP/IP LAN. Effective camera move control (final stage of the IFS, on PC4) was not implemented since it depends only on the quality of the aggreagtion result.

A. Experiments

The first experiment allows us to check if the sound system (A in Figure 1) correctly updates the direction of the sound source when this one moves. We have

checked that the import function is able to read information produced by the signal processing algorithm of the sound capture system, up to 5 times per second. The IFS system could accept a higher refresh rate but this will be useless due to the slow moving rate of the speaker. Tests simply dump sound direction and user's identity into a file.

The second experiment validates the video system (B in Figure 1) in the same way as the sound system. PC2 has to import data, apply the grouping algorithm and compute the COG of user's heads up to 5 times per second. The same remark holds for what concerns the refresh rate. We noted that the coordinates of markers are correctly imported and that the algorithms give correct positions for the COG of each head, *even when the number of users changes* (we only suppose that each user's head is equipped of two markers).

The third experiment checks information produced by the aggregation function and verifies that the result is updated when the speaker moves. In contrast with the two previous cases, this experiment involves the three computers and the LAN (see below for implementation details). We checked that data communications between A, B and C are correctly synchronized without data loss or bad interpretation. We also checked that speaker's moves generate a new selection sequence with data correctly reflecting these moves. Finally, *on the fly* modifications of the fusion function parameters (either D_p, I_p or n_r) are effectively taken into account by the running IFS and produce expected variation of the fusion result.

B. Implementation

Our implementation of the video conference room IFS is a first step in design and implementation of a generic adaptable and controlled distributed IFS (ACDIFS). In a previous work [BHMP07], we implemented an IFS on top of Global Sensor Network (GSN) [AHS07]. However, restricted capabilities of GSN with respect to run-time adaptation for fusion functions parameters and modification of the fusion functions network lead us to develop a new system based on interconnected OSGi [OSG] plateforms. An OSGi plateform is a Java based middleware hosting services grouped into *bundles*. Bundles may be inserted/removed in the plateform at run-time and services related to the moved bundle may be notified.

In our ACDIFS, fusion functions are implemented as Java classes. Interconnections of the fusion graph and parameters of the fusion functions are defined in XML files (at least one per computer running a part of the IFS). In the initial phase, the set of OSGi plateforms are first started. Then, bundles of the fusion functions are installed by the plateforms using the XML IFS description files. Finally, the fusion functions are started by each OSGi plateform. Updating function parameters in the XML files generates method calls in the Java fusion classes to update corresponding internal variables. Details of our implementation will be presented in a forthcomming report.

VI. Conclusion

We have presented a video conference room system design and implementation. It is based on two capture systems, respectively for sound signal processing and for image processing. The sound system manages directions and recognition of the speaker whereas the video system isolates his position from those of the participants. These two kinds of information are fused to control a tarcking video camera. The quality of the fusion functions are controlled by parameters set at run-time.

This system is analyzed and designed as an Information Fusion System (IFS). Our approach is first to describe the IFS as a data-flow graph where each node represents a fusion function. Fusion functions are connected through 3 groups of ports. In a second step, we map fusion fucntions and links between functions to software units. In this paper, we have developped a mapping on interconnected OSGi frameworks.

The resulting system behaves as expected. Due to slow movements of people, low sampling rates of the sound and the video systems may be used.

Future work will follows two directions. The first one relates to the fusion algorithms for the sound and the video systems. We plan for instance to introduce several possible speakers and not only one.

The second direction will study applications and extensions of our Distributed IFS. On one hand, we will apply our framework to other D-IFS, like power systems, building management, etc. On another hand, we will introduce automated control of reconfiguration of a running D-IFS, in contrast to present operator or designer based reconfiguration.

References

[AHS07] Aberer Karl, Hauswirth Manfred, and Salehi Ali. Infrastructure for data processing in large-scale interconnected sensor networks. In *Mobile Data Management (MDM)*, 2007.

[BHMP07] Benoit Eric, Huget Marc-Philippe, Moreaux Patrice, and Passalacqua Olivier. Integrating OPC data into GSN Infrastructure. Technical report, LISTIC - University of Savoie, 1 October 2007.

[FKI+04] F. Asano, K. Yamamoto, I. Hara, J. Ogata, T. Yoshimura, Y. Motomura, N. Ichimura, and H. Asoh. Detection and Separation of Speech Event Using Audio and Video Information Fusion and Its Application to Robust Speech Interface. *EURASIP Journal on Applied Signal Processing*, pages 1727–1738, November 2004.

[KH07] Khairunizam WAN and Hideyuki SAWADA. 3D Measurement of human upper body for gesture recognition. In *International symposium on Optomechatronics Technology (ISOT2007)*, volume 1.6718, 1-8, 2007.

[Kno] Knopflerfish project. Knopflerfish - Open Source OSGi. *http://www.knopflerfish.org/*.

[OSG] OSGi Alliance. official web site. http://www.osgi.org/.

[TKTH04] Toshiya TAKECHI, Koichi SUGIMOTO, Takashi MANDONO, and Hideyuki SAWADA. Automobile identification based on the measurment of car sounds. *Annual Conference of IEEE Industrial Electronics Society, TD6-4*, 2004.