REASONING FOR VIDEO-MEDIATED GROUP COMMUNICATION

Manolis Falelakis[1], *Rene Kaiser*[2], *Wolfgang Weiss*[2] *and Marian F. Ursu*[1]

[1]Department of Computing
Goldsmiths, University of London
London, UK
[2]Institute of Information and Communication Technologies
JOANNEUM RESEARCH
Graz, Austria
m.falelakis@gold.ac.uk, {rene.kaiser, wolfgang.weiss}@joanneum.at, m.ursu@gold.ac.uk

ABSTRACT

In this paper we present an approach to the reasoning required to support multi-location, multi-camera group-to-group video communication, which we call *orchestration*. Orchestration is akin to virtual directing: it has to ensure that each location displays the most adequate shots from all the other available sources. Its input is low-level cues extracted automatically from the AV streams. They are processed to detect higher-level events that determine the state of the communication. Directorial decisions are then inferred, reflecting social communication as well as stylistic criteria. Finally, they are transformed into camera and editing commands, directly executable by the AV infrastructure. Here, we present the architecture of the Orchestrator and sketch our rule-based approach to reasoning.

Index Terms— virtual director, orchestration, event processing, rule-based reasoning

1. INTRODUCTION

We investigate video-mediated group-to-group communication between distant locations equipped with multiple cameras. Compared to 1-to-1 video chats, the number of video streams available in such setups is much larger and displaying all available views as a *mosaic* on the same screen is not an option. High-resolution images of a certain size are a necessary quality to effectively transport non-verbal messages (mimics, gestures) that are essential to human communication. In a videoconference-like setup with a considerable number of participants, intelligent software is required to automatically identify which of the many actions or events currently happening is most relevant for each individual participant so that the most appropriate camera views can be continuously selected (cf. principles presented in [1]). We refer to such a software module as *Orchestrator*.

The concept of orchestration is akin to *virtual directing*; it is similar to the decisions taken by directors, cameramen, and editors when composing programmes recounting live events [2]. Contrary to conventional film production where a single narrative thread is compiled, the Orchestrator has to reason for each individual screen and take the events from all the participating locations into account. The participants are at the same time viewers and actors and, furthermore, the Orchestrator itself plays an active part in the interaction.

Beside the main pragmatic goal of covering the conversation with appropriate camera views, the Orchestrator also aims to enhance the immersiveness of the experience. All in all, it aims to create a communication medium as natural and immersive as possible.

In this paper we introduce the concept of *communication orchestration* from a technical standpoint. We present a declarative approach to reasoning and sketch the components of a corresponding architecture, namely the one we implemented. These are the main contributions of the paper, as video-mediated communication *orchestration* has not been defined in technical terms elsewhere.

2. ORCHESTRATION LOGIC

The Orchestrator is designed to receive low-level cues from an AV analysis component (see [3]) and to output camera directives and screen editing commands. Its position within the overall communication system is depicted in Figure 1.

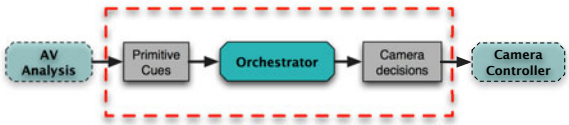

Fig. 1. The system boundaries of the Orchestrator

Table 1. Example of editing rules in a pseudo-notation resembling FOL

1.	$turnShift(P) \wedge isInOtherLocation(L, P) \rightarrow L : cut2CUFront(P)$
2.	$crossTalk(P) \wedge isInOtherLocation(L, P) \rightarrow L : cutAway2CUFront(P)$
3.	$relevancyMarker(P) \wedge isInOtherLocation(L, P) \rightarrow L : cut2CUFront(P)$

2.1. Desired behaviour

We have developed the concept of *Orchestration* in collaboration with experts from the fields of social interaction and cinematography. In order to define the behaviour of the Orchestrator, we undertook a knowledge elicitation process that resulted in the semi-formal statement of a set of rules, illustrated in Table 1. They can be roughly divided into two categories, driven by notions of (i) pragmatics of social communication and (ii) style of visual narration.

Rules of the former (pragmatic) category ensure that the most important aspects of the communication are well depicted at each location. For instance, rule 1 of Table 1 states that "if it is person's P turn, say, in a social game, and P is not within location L, then the screen in location L should display a close-up front shot of P". Rule 3 is another example from this category and states that "when a conversation relevancy marker is spotted from person P then the screen of location L should display a close-up front shot on P". As relevancy markers we consider the words "so", "well" and "but", when they occur within the first four words of a turn shift as identified in [4].

Rules of the second (stylistic) category impose aesthetic principles learned from cinematography to the communication coverage. Rule 2, for example, states that "if there is a cross-talk from person P, the screen of L should cut-away (display for some time and then return to previous shot) to a close-up front of P". A cross-talk is defined when a person P starts talking but this does not result in a turn shift of the conversation (informally, P is talking "over" an existing conversation).

2.2. Issues of the reasoning mechanism

For the reasoning mechanism to be able to execute rules as the ones described above, there are a number of issues that need to be tackled:

1. The conditions of the rules are expressed using entities/events (e.g. "turnShift" or "crossTalk") that are not directly detected by the AV Analysis component. Primitive cues emerging from the latter have to be "lifted" to a higher semantic level to infer such predicates.

2. Desired camera selection and screen editing behaviour needs to be formalized in a way that it is both executable by the Orchestrator and flexible enough to be amended and fine-tuned by humans. Moreover, synchronous triggering of multiple rules may produce contradicting results, making a conflict-resolution mechanism necessary.

3. The rules refined by experts are expressed in terms of *shots* (e.g. "close-up front of person P"). These ought to be translated into camera directives on the basis of the dynamic position of the participants and spatial characteristics of the cameras (orientation, lens angle, focal distance).

4. The interaction largely depends on the current state/context of the conversation. The same events can trigger different responses if they occur in different contexts. This notion naturally calls for modelling the Orchestrator partly as a state-machine.

In the following section we present an architecture that facilitates the automatic application of the rules while efficiently addressing the aforementioned issues.

3. ARCHITECTURE

As discussed, the Orchestrator processes low-level cue streams to produce directives about what to be displayed on each screen. The event-driven nature of this process, together with the needs for temporal reasoning and real-time processing led us to the decision of implementing the Orchestrator using a rule engine enabled for event-processing (cf. [5]).

The architecture of the Orchestrator is illustrated in Figure 2. Here we briefly describe the basic modules that constitute it, namely the Cue Lifter, Geometric Reasoner, Directors, Camera Compilers and the associated Knowledge Base.

3.1. Cue Lifter

The Cue Lifter tackles issue 1 by lifting primitive cues received from AV Analysis to a more abstract, higher semantic level. It generates semantically richer events such as "turnShift" and "crossTalk" based on definitions expressed declaratively as rules, as illustrated in Table 2. An overview of the cues the Orchestrator operates with is given in [3]. For example, the first rule states that if "voice activity" is detected for person P and P has not been talking, i.e. it was someone else's turn, and P is not cross talking, then a "turnShift" event is generated for P - P takes the turn of the conversation. The second example rule defines the "cross talk", where $\mathbf{d}$ denotes the

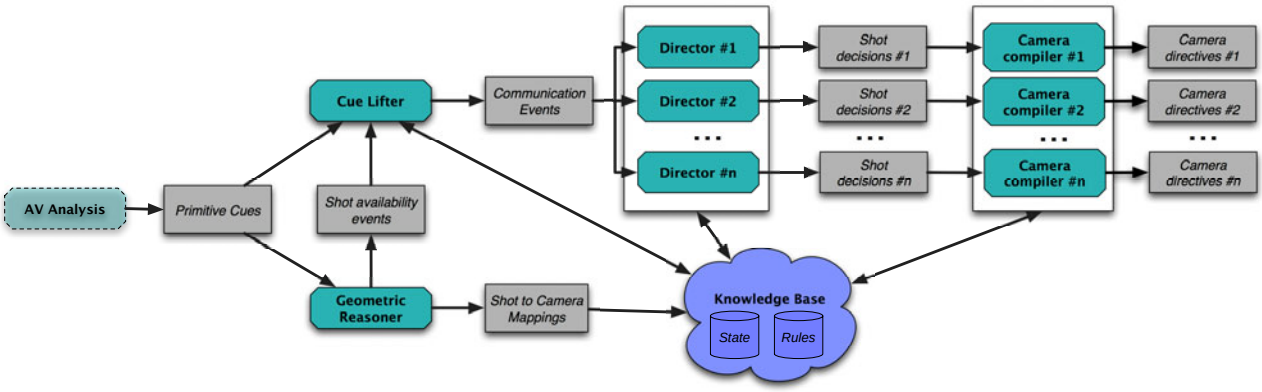

Fig. 2. Orchestrator architecture outline. Directors and Camera Compilers operate per location.

temporal operator "during" as used in Allen's interval algebra [6]. A "relevancyMarkerSpotted" event of person P with the word W indicates an important word (such as "so", "well" or "but") close the beginning of the turn of P.

Before we moved to a rule-based approach for cue lifting, we investigated the applicability of description logics based on an OWL model and Jena rules[1] with built-in SPARQL queries. Although not necessarily inherent to the approach itself, lifting cues using an ontology resulted in a scaling problem [7]. Early evaluations of our latest rule-based implementation using JBoss Drools[2] couldn't identify such performance bottlenecks. For a general discussion of video event understanding, we refer to the overview presented in [8].

3.2. Director

The application of the editing rules, i.e., the ones that make shot decisions for each particular screen (such as those presented in Table 1) is a task undertaken by the Director component. In our architecture (Figure 2), there are multiple Director instances running in parallel: there is one Director instance running per location, but they do not operate in isolation. Directors receive high-level events from the Cue Lifter originating from all the locations and query the Knowledge Base to ensure their awareness of the current global state of the interaction before they apply rules. The input is global, but the conclusions are local. We chose to model the problem this way because it resembles the way human editors operate.

3.3. Geometric Reasoner and Camera Compiler

As identified as issue 3 of section 2.2, decisions about shots need to be compiled into camera directives. The Geometric Reasoner and Camera Compiler modules carry out this task.

The Geometric Reasoner continuously computes the degree up to which each camera supports each one of the possible types of shots for each person, as illustrated in Figure 3. Calculations are based on information regarding the overall setup, angles and characteristics of cameras combined with the position and head orientation of each person. The shot types considered in our approach are listed in Table 3.

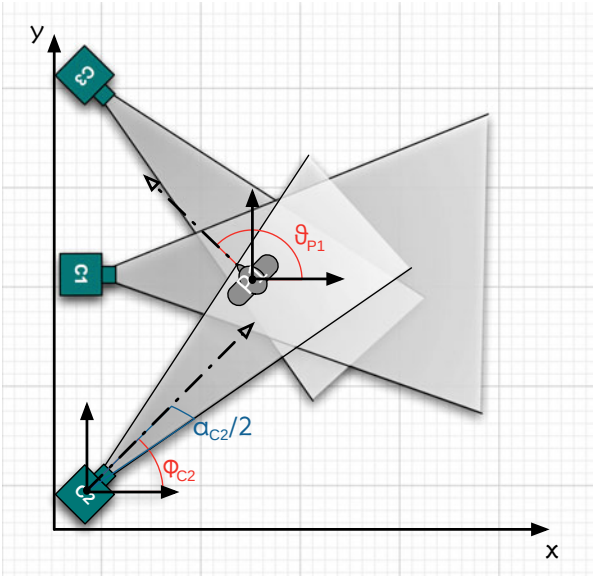

Fig. 3. Geometric Reasoner: Computing the support each camera provides to each shot type for a person in order to resolve shot decisions to camera directives.

The Geometric Reasoner outputs (i) an event stream that notifies the Directors when a shot is or is not currently attainable and (ii) a mapping for all the possible shots within a location to corresponding camera directives, together with associated confidence factors. The Geometric Reasoner is taking the participants' head orientation into account to determine how well a shot is realized by a certain camera.

Whenever a shot decision is made, the mapping is read

[1] http://jena.sourceforge.net/inference/
[2] http://www.jboss.org/drools

Table 2. List of cue lifting rules

1.	$voiceActivity(P) \wedge \neg turn(P) \wedge \neg crossTalking(P) \rightarrow turnShift(P)$
2.	$voiceActivity(P1)$ **d** $voiceActivity(P2) \wedge \neg turn(P1) \rightarrow crossTalk(P1, P2)$
3.	$turnShift(P, T1) \wedge keywordSpotted(W, P, T2) \wedge T2 - T1 < 1sec \rightarrow relevancyMarkerSpotted(P, W)$

Table 3. Shot types at a particular location

Shot name	Description
CUFront(P)	Close-up front shot on person P
CUObliqueLeft(P)	Close-up oblique shot on P from the right side
CUObliqueRight(P)	Close-up oblique shot on P from the left side
MidShot(P)	Frontal Mid shot on P
Wide	Wide shot from frontal camera

from the Knowledge Base by the associated Camera Compiler to translate the shot into an actual camera decision.

3.4. Knowledge Base

Addressing the fourth issue, the Knowledge Base is shared between all the reasoning modules (as depicted in Figure 2). It manages both global and location-specific states and offers an interface for efficient manipulation and querying.

4. CONCLUSIONS

We described a declarative approach to the automatic reasoning that supports the orchestration of multiple cameras in a multi-person, multi-location videoconferencing system. This is a novel concept and is accompanied by a concrete implementation.

Low-level AV cues are processed and lifted to a higher semantic level, as required by the rules expressing the shot selection process. These rules embed both pragmatic criteria, which ensure that the essential communication messages, verbal or non-verbal, are conveyed, and stylistic criteria, which aim to enhance the immersiveness of the communication. We presented the orchestration concept, main issues that need to be tackled for automatic reasoning, and an architecture we implemented that addresses them. Preliminary performance evaluations with example rules have shown that the processing time is both short (below 1ms) and constant.

We are currently extending the rules set to work with fuzzy and probabilistic knowledge: i.e. to reason with cues with estimated confidence values. We are further dealing with inconsistent rules, as resulting from elicitation from experts, and plan to investigate deontic logics. Another extension, at this end, is the ability to achieve prediction behaviour, i.e. to detect events that are likely to happen with good enough confidence.

Acknowledgements

This work was performed within the Integrated Project TA2, Together Anytime, Together Anywhere[3]. TA2 receives funding from the European Commission under the EU's Seventh Framework Programme, grant agreement number 214793.

5. REFERENCES

[1] Marc Christie, Rumesh Machap, Jean marie Norm, Patrick Olivier, and Jonathan Pickering, "Virtual camera planning: A survey," in *In Proceedings Smart Graphics*. 2005, pp. 40–52, Springer.

[2] Doug Williams, Marian F. Ursu, Joshan Meenowa, Pablo Cesar, Ian Kegel, and Karl Bergstrm, "Video mediated social interaction between groups: System requirements and technology challenges," *Telematics and Informatics*, vol. In Press, 2010.

[3] Danil Korchagin, Petr Motlicek, Stefan Duffner, and Hervé Bourlard, "Just-in-time multimodal association and fusion from home entertainment," in *Proceedings IEEE International Conference on Multimedia & Expo (ICME), Barcelona, Spain*. 2011, Springer.

[4] Martin Groen, Jan Noyes, and Frans Verstraten, "The effect of substituting discourse markers on their role in dialogue," in *Discourse Processes*, 2010, vol. 47, pp. 388–420.

[5] Opher Etzion and Peter Niblett, *Event Processing in Action*, Manning Publications Co., 2010.

[6] James F. Allen, "Maintaining knowledge about temporal intervals," *Commun. ACM*, vol. 26, pp. 832–843, 1983.

[7] Rene Kaiser, Claudia Wagner, Martin Hoeffernig, and Harald Mayer, "The interaction ontology model: supporting the virtual director orchestrating real-time group interaction," in *Proceedings of the 17th international conference on Advances in multimedia modeling*, 2011, MMM'11, pp. 263–273.

[8] Gal Lavee, Ehud Rivlin, and Michael Rudzsky, "Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video," *Trans. Sys. Man Cyber Part C*, vol. 39, no. 5, pp. 489–504, 2009.

[3] http://ta2-project.eu/