A Specification Language for Dynamic Virtual Video Sequence Generation

Craig Lindley, Anne-Marie Vercoustre

A Specification Language for Dynamic Virtual Video Sequence Generation

Craig A. Lindley,
CSIRO Mathematical and Information Sciences,
Locked Bag 17, North Ryde NSW 2113, Australia,
and
Anne-Marie Vercoustre,
INRIA-Rocquencourt, B.P. 105-78153 Le Chesnay Cedex, France

The FRAMES project is developing a system for video database search, content-based retrieval, and virtual video program synthesis. For dynamic synthesis applications, a video program is specified at a high level using a virtual video prescription. The prescription is a document expressed in a semi-formal language specifying the video structure, including formal specifications for direct reference to specific video components, components specified and retrieved by parametric query, and components generated as associative chains based upon an initial specification. This paper describes the syntax and semantics of the specification language for associative chaining. The language allows the specification of entity types for association generation, the specification of conjunctions and disjunctions of entity types, initial values, fixed constraints, and weights upon entity types. A short example is presented to illustrate the use of the language and how it may be used to generate different video sequences from a common database of video components.

Keywords: video synthesis, associative reasoning, video semantics

Introduction

The FRAMES project is developing a system for video database search, content-based retrieval, and virtual video program synthesis. Video components within the FRAMES database are described in terms of a multi-layered model of film semantics, derived from film semiotics. For dynamic video program synthesis applications, a program is specified at a high level using a virtual video prescription [6]. The prescription is a document specifying the video structure, with specific segments specified by direct reference to video components, and dynamically selected segments specified by parametric queries. For video synthesis applications, coherent sequences of video are required, rather than just lists of material satisfying a common description. To meet this requirement, the FRAMES system uses an *association engine* for generating associative chains of video sequences, initiated by an initial specification embedded within a virtual video prescription. Sequences generated from an initial specification include elements that are approximately matched to the specification, and the criteria for matching evolve dynamically as the sequence develops. This paper describes the specification language for associative chaining. We begin with an overview of the multi-layered semantic model. FRAMES video search processes are then summarized, including the operation of the association engine. The syntax of the associative chaining specification language is then described. A short example is presented to illustrate the use of the language and how video sequences are generated. The semantics of the language are then described in more detail.

Video Semantics Metamodel

A video semantics metamodel is the core component of the FRAMES system [5]. This model provides a set of entity and relationship types for expressing aspects of the *meaning* of a video segment that may be perceived or read into the segment by a viewer at a number of levels of perceptual discrimination and abstraction. Based upon the film semiotics pioneered by the film theorist Christian Metz (1974), we identify five levels of cinematic codification that are represented within the metamodel:

1. the *perceptual level*: the level at which visual phenomena become perceptually meaningful, the level at which distinctions are perceived by a viewer within the perceptual object. This level includes perceptible visual characteristics, such as colours and textures. This level is the subject of a large amount of current research on video content-based retrieval (see [1]).

2. the *diegetic level*: at this level the basic perceptual features of an image are organised into the four-dimensional spatio-temporal world posited by a video image or sequence of video images, including the spatiotemporal descriptions of agents, objects, actions, and events that take place within that world. This is the "highest" level of video semantics that

most research to date has attempted to address, other than by associating video material with unconstrained text (allowing video to be searched indirectly via text retrieval methods, eg. [9]).

3. the *cinematic level*: the specifics of formal film and video techniques incorporated in the production of expressive artefacts (“a film”, or “a video”). This level includes camera operations (pan, tilt, zoom), lighting schemes, and optical effects. Automated detection of cinematic features is another area of vigorous current research activity (see [1]).

4. the *connotative level*: this is the level of metaphorical, analogical, and associative meanings that the denoted (ie. diegetic) objects and events of a video may have. The connotative level captures the codes that define the culture of a social group and are considered “natural” within the group.

5. the *subtextual level*: this is the level of more specialised meanings of symbols and signifiers. Examples might include feminist analyses of the power relationships between characters, or a Jungian analysis of particular characters as representing specific cultural archetypes.

The connotative and subtextual levels of video semantics have generally been ignored in attempts to represent video semantics to date, despite being a major concern for filmmakers and critics. Modelling “the meaning” of a video, shot, or sequence requires the description of the video object at any or all of the levels described above. The different levels interact, so that, for example, particular cinematic devices can be used to create different connotations or subtextual meanings while dealing with similar diegetic material.

Metamodel components are drawn upon in the FRAMES authoring environment in order to create specific models, or *interpretations*, of specific video segments. There may be more than one model at any particular level, for example, corresponding to different theories of subtext or created by different people. Cinematic and perceptual level descriptions may be generated automatically to an increasing extent. Subtextual and connotative descriptions are necessarily created by hand. The diegetic level represents an interface between what may be detected automatically and what must be defined manually, with ongoing research addressing the further automation of diegetic modelling (eg. [4]).

FRAMES Video Processing

Virtual videos can be specified by an author in the form of virtual video prescriptions [6] based on the concept of virtual document prescriptions [10]. The first version of the FRAMES dynamic virtual video synthesis engine, currently under development, will support three methods by which video segments may be retrieved and inserted within a dynamic virtual video stream, according to initial specifications expressed in a virtual video prescription.

Access by direct reference uses an explicit, hard-coded reference to a video data file plus start and finish offsets of the required segment (eg. using the referencing syntax of SMIL [3]). While this is a simple, fast, direct, and specific method of accessing video data, it requires the author of a prescription to have knowledge of the exact contents of the data, the data must not change, and it is not robust against changes in the location of the data. This method also does not support dynamic reselection of data based upon parameters passed into a virtual video prescription.

Access by parametric match overcomes these deficiencies by allowing video components to be retrieved if their descriptors match conventional database queries (eg. expressed in SQL). Database queries may contain complex logical and pattern matching operations.

In parametric search, the initial query may form a hard constraint upon the material that is returned, such that all of its conditions must be satisfied. Alternatively, a *ranked parametric search* can return a list of items ranked in decreasing order of match to the initial query, down to some specified matching threshold. Matching is conducted against the number of conditions expressed in the query, and the number of conditions that are satisfied by a component provides the basis for its ranking. This algorithm is different from other fuzzy retrieval algorithms for images, where the fuzziness is based upon attribute values (eg. a selection on color="red" will weight decreasingly values "red", "red-orange", and "orange", or use the probability that the attribute value in the database is "red").

Access by associative chaining is a less constrained form of accessing video data, where the material incorporated into a dynamically generated video sequence consists firstly of the component that has the strongest match to an initial search specification, and then includes components that incrementally match an evolving specification. Associative chaining starts with specific parameters that are progressively substituted as the chain develops. At each step of associative chaining, the video component selected for presentation at the next step is the component having descriptors that most match the initial specification, parameterised using values from the descriptors attached to the video segment that is presented at the current step. Since association is conducted progressively against descriptors associated with each successive video component, paths may follow semantic chains that progressively deviate from the initial matching criteria. In effect this is a cyclic ranked parametric query strategy. In this strategy the descriptor values for the current video component are used to parameterise a ranked query that selects the next video component to be displayed in the dynamically generated sequence; the next component then becomes the current component, and the matching process using ranked matching is repeated. Termination occurs when a termination condition specified in the initial associative chaining specification is satisfied, or when no more sequences match the chaining specification.

Association Specification Language

The video semantics metamodel provides a set of component and relationship types that may be instantiated to model the meaning of video segments within a video database. The associative chaining algorithm summarised above provides the method of associating one segment with another in order to create a meaningful sequence. The *association specification language* provides a syntax for describing which entity and relationship types the associative chaining algorithm is to take into account in generating associative chains, and how to take them into account. Specifications expressed in this syntax include:

- termination conditions for a sequence, in terms of number of segments (n), overall play length (time), or a minimum matching threshold (rank)
- semantic model components, including objects, attributes, and relationships, that are to be matched in order to create the associations between video segments within an associative chain
- initial values for semantic model objects, attributes, and relationships at the start of an associative chain
- constraints upon the values of semantic model objects, attributes, and relationships that are fixed through an associative chain
- *weightings* upon semantic model objects, attributes, and relationships
- propositional operators within or between matching criteria: AND, OR, and NOT
- variables for passing values into associative chaining specifications within a prescription

Hence this language allows the author of a virtual video prescription to express the criteria that will be used by the associative chaining algorithm for the selection of the next most highly associated component at each step in the generation of an associative chain. In this way, the language expresses aspects of the semantics of the video material as read by the author of it's descriptive annotations, and the semantics of how those descriptors are to be processed by the chaining algorithm. The BNF for the associative chaining language is presented in the Appendix to this paper.

Association specifications form a particular query type embedded within a virtual video prescription. Direct reference and unranked parametric queries represent different query types within a prescription, and are therefore not addressed by the association specification language. The association specification language can be used to specify single ranked queries, in which case the associative chaining algorithm is invoked with the specified termination condition set at a single returned segment (ie. terminate when n > 1).

Example

As an example of how the associative matching algorithm is triggered and proceeds from an initial specification, we assume a very simple database containing four video components, two each for the characters Freda and Fred, and one each representing each of those characters in Sydney and Paris. The four components are represented on Figure 1 in terms of these descriptor values.

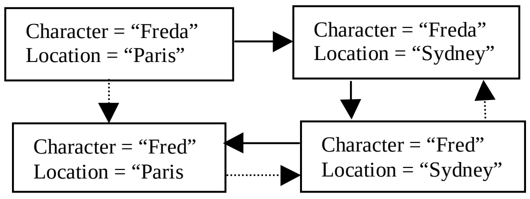

Figure 1. Descriptors representing four example video components, with association paths between those components.

Suppose a virtual video prescription contains the associative chaining specification: "assoc_match: (Character[w:0.5] = "Freda", Location = "Paris") terminate(n>10) ?end". Chaining will begin with the video segment depicting Freda in Paris. The explicit weighting on Character means that it is weighted higher than the default, so the next element in this associative chain will be the one having the same character, but a different location. After that, there are no other components with Character = "Freda" that have not been displayed in this chain, so the next match will be on Location = "Sydney". That match replaces the current character value with "Fred", so the next match from that point is another match on Fred, but this time with Location = "Paris". The resulting associative chain is that depicted by the unbroken arrow. A similar specification, but with a higher weighting attached to Location instead of Character, will result in the very different associative chain depicted by the broken arrows.

Specification Language Semantics

Specified entity, attribute, and relationship types are matched with those associated with video sequences within each step of the associative chaining process. The match of specified types and their values with those in descriptors is the basis for ranking the degree of association of sequences with the specification and its currently instantiated values. Logical and relational operators between terms in a specification, in order of precedence and with ranking implications, include:

Logical Operator	Boolean Meaning	Ranking Meaning if Satisfied
!	NOT	x 1
!=	NOT EQUIVALENT	x 1
= =	EQUIVALENT	x 1
,	AND	Σ
;	OR	Greatest Disjunct

A weight immediately following an entity type weights all video sequences described as having entities of that type. Weights associated with video segments that satisfy more than one term of a conjunctive expression are summed across each satisfied term. When a video segment satisfies more than one term of a disjunctive expression, the weight assigned to the most heavily weighted term is used to calculate the weight of the sequence. These semantics applied to primitive logical expressions are inherited through any higher levels of nesting. When the terms of a disjunction have different signs, the total weight of a sequence is calculated using the maximum weight of the positive disjuncts that it satisfies, minus the greatest absolute value among the negative weights of disjuncts that it satisfies.

Within a specification, a single “=” signs denotes an initial matching value for an entity type. Double “= =” or “!=” operators indicate constant constraints (ie. do not vary between matching steps). In general, constraints may be *hard* (ie. *must* be satisfied throughout an associative chain) or *soft* (ie. are *preferred*, but not necessary, values). Here the hard constraint is interpreted to mean that a constrained descriptive term cannot have any other value, unless it occurs elsewhere within a specification. (An alternative interpretation, not currently used in FRAMES, would be that all sequences *must* satisfy this condition, which removes it from the ranking calculation.) In the associative chaining process, a soft constraint amounts to a preferential weighting of a preferred term or value. To keep the specification language simple, soft constraints do not have a dedicated lexical symbol, but can be expressed as a disjunctive condition, with the same semantic component in each disjuct, but having one disjunct weighted and a hard constraint specified upon its value. For example, the expression “(A ; A[w:x] == “<*constant*>”) applies the preferential weight x to A when it has the value “<*constant*>”; when A has some other value, it is still used for matching, but without the explicit weight. Both disjuncts could be weighted, in which case the preference goes to the disjunct with the highest weight.

Related Work

The associative chaining approach described in this paper is based upon the Automatist storytelling system developed at MIT and demonstrated in the ConText [2] and the ConTour [8] systems for generating dynamic evolving documentaries from a changing database of video material. The Automatist system uses simple keyword descriptors specified by authors and associated with relatively “self-contained” video segments. The FRAMES system extends this basic associative chaining approach by using the highly structured semantic model described above, which allows much greater discrimination on descriptor types, and more specific forms of relationship between sequenced video components. FRAMES also goes beyond the Automatist system by introducing the flexible associative chain specifications expressed using the language described in this paper, the inclusion of direct reference and parametric retrieval of video components, and the specification of virtual videos using the virtual video prescription mechanism.

Conclusion

The FRAMES association specification language is a flexible and convenient language for the specification of the terms and constraints upon dynamically generated video sequences. Specific filmic structures and forms can be generated in FRAMES by using particular description structures, association criteria and constraints expressed in this language. In this way the sequencing mechanisms remain generic, with emphasis shifting to the authoring of specific models, interpretations, and specifications for the creation of specific types of dynamic virtual video productions. The identification of the particular types of descriptors and constraints appropriate for the generation of different genres and forms of video program is an important topic of ongoing research.

References

[1] P. Aigrain, H. Zhang, and D. Petkovic, “Content-Based Representation and Retrieval of Visual Media: A State-of-the-Art Review”, Multimedia Tools and Applications, vol. 3, pp. 179-202, Klewer Academic Publishers, The Netherlands, 1996.

[2] G. Davenport and M. Murtaugh, “ConText: Towards the Evolving Documentary”, Proceedings, ACM Multimedia, San Francisco, California, Nov. 5-11, 1995.

[3] P. Hoschka (ed), “Synchronised Multimedia Integration Language”, W3C Working Draft 2-February, (work in progress), 1998.

[4] M. Kim, J. G. Choi, and M. H. Lee 1998 “Localising Moving Objects in Image Sequences Using a Statistical Hypothesis Test”, Proceedings of the International

Conference on Computational Intelligence and Multimedia Applications, pp. 836-841, 1998.

[5] C. A. Lindley, “A Multiple-Interpretation Framework for Modeling Video Semantics”, ER-97 Workshop on Conceptual Modeling in Multimedia Information Seeking, LA, 6-7 Nov., 1997.

[6] C. A. Lindley and A.-M. Vercoustre, “Intelligent Video Synthesis Using Virtual Video Prescriptions”, Proceedings of the International Conference on Computational Intelligence and Multimedia Applications, pp. 661-666, 1998.

[7] C. Metz, *Film Language: A Semiotics of the Cinema*, trans. by M. Taylor, The University of Chicago Press, 1974.

[8] M. Murtaugh, *The Automatist Storytelling System*, Masters Thesis, MIT Media Lab, Massachusetts Institute of Technology, 1996.

[9] U. Srinivasan, L. Gu, K. Tsui, and W. G. Simpson-Young, “A Data Model to Support Content-Based Search in Digital Videos”, *Australian Computing Journal*, vol. 29, no. 4, pp. 141-147, 1997.

[10] A.-M. Vercoustre and F. Paradis, “A Descriptive Language for Information Object Reuse through Virtual Documents”, Proceedings of the 4th International Conference on Object-Oriented Information Systems (OOIS'97), pp. 299-311, 1997.

Appendix: BNF For the FRAMES Association Specification Language

The syntax for specifying associative chains is described by the following BNF definition:

```
assoc_match     ::= "assoc_match"  ": " condition_list
                        [ terminate ] “?end”
condition_list  ::=  AND_list
AND-list        ::=  OR_list  {  ","  OR_list  }*
OR_list         ::=  cond  {  ";"  cond  }*
cond            ::=   simple_cond  |   neg_cond  |
                        "(" condition_list ")"
neg_cond        ::=    "!"  simple_cond |
                        “(“ condition_list “)”
simple_cond     ::=  %name [ weight ] [ initial_value |
                constant_value | neg_cst_value]
weight          ::=  "[w: "  %value  "]"
initial_value   ::=   "="    %string
constant value  ::=   "=="  %string
neg_cst_value   ::== "!="  %string
terminate       ::= "terminate" "("  rank_cond   ")"
rank_cond       ::=  [ "rank"  "<"   %value ]  nb_cond
nb_cond         ::=  [ "n"       ">"    %value]
                        [ time_cond ]
time_cond       ::= "time" ">" %value
```

Statements are enclosed in parentheses, parentheses may be nested, inner nested statements have ranking evaluation precedence over enclosing statements, and weights are indicated within square brackets.