**BIDDALA — Hypergraph**

<3 figures>

**Hypergraph-Based Collaborative Film Archive**

Biddala, S. R. and Singh, N.

Digital film archives in addition to preserving cinema should also accommodate and provide for the computation of its cinemas by means of the metadata provided to represent the content in them. The provision for the computation of cinema could lead to various applications like semantic search, automatic genre clustering and classification, special visualization tools, and robust comparisons of movies.

This paper describes a data model for storing the metadata of a film in an archive based on a discrete theory of cinema that has been in development for the past four years based on Indic theories and other theories of drama and film (Muni, 1996). This theory reduces cinema to a hypergraph composed of tags that assume its full form only with collaborative tagging and extensive visual and textual computing. A system called CinemaScope is also being developed based on this data model which uses HypergraphDB as its database and is being designed and developed to be a semiautomated annotation database system.

# **Overview of Theory**

Cinema is ontologically a three-tier structure:

1. Spatio-temporal flow controlled play of light and sound.

2. Mere-temporal ‘flow’, which is visually created by a play of 25 fps.

3. Trans-temporal content, which is embedded in a ‘flow’.

Cinema concludes in a thematic unity. Unit of a discrete contiguum of cinema is punctuation. Idea of punctuation is inspired by Leibnizian distinction between ideal point and actual point on the one hand and continuum and discrete contiguum on the other (Leibniz, 2013). Punctuation is a form of an actual point that tells apart two recognizable and non-identical discernible contents. It has a structure <entity1|entity2, relational context> or graphically <node1|node2, edge>.

There are three classes of punctuations based on the three tiers of ontology of cinema:

1. Temporal punctuation.

2. Vectorial punctuations between embedded content.

3. Mereological punctuations between vectorial contents and the thematic conclusion of cinema.

This way the form of punctuation constitutes discrete contiguum of cinema, which is computationally reducible to a giant graph that tells truth about cinema as art.

Ontologically there are categories of discrete content that are embedded in punctuated units of flow like shot and episodes. Some of the categories form the content part of the cinematic narrative (like locale, characters, events, and actions) while the rest (camera work, editing, re-recording) form the expressive part of it (Chatman, 1978). Forces operative on these entities as vector punctuations create kinetics of cinema toward thematic conclusion. The motion in films is through the movement of story or plot from the beginning to the conclusion. Such a motion towards theme is punctuated by points of events and points of considerations. It is through the sequence of events that connection to the theme is established. These events are punctuated with points of considerations. Story can be rendered as sequence of the points of events, and under each event there are several considerations (Dhananjaya, 2004). There are scalars between the sequences of cinema that are merely informative and do not account for the motion of the plot.

# **Overview of Data Model**

In Hochin (2006) and Hochin and Tatsuo (2000), a data model called Directed Recursive Hypergraph Data Model (DRHM) has been described in which the content of multimedia is reduced to nodes and edges of a hypergraph. In Radev et al. (1999a; 199b), the structural and behavioral aspects of data that form multimedia information systems have been modelled as a graph-based object-oriented model, and a possible data model for film is shown. These approaches suggest that a graph-based data model for cinema based on the discrete theory of cinema proposed in the previous section would best represent it. Punctuation or point <node | node, edge> can be seen as triple of tags. This will make possible computing of the discrete contiguum of cinema as a graph. Also there are complex relations between the various entities of cinema which is best represented by hyperedges.

The graph data model should accommodate the various ontological entities and their punctuations, the mereological punctuations as well as the vectors between episodes and the considerations between events. The partial data model for the cinematic hypergraph described is represented in the form of Venn diagram as shown in Figure 1.

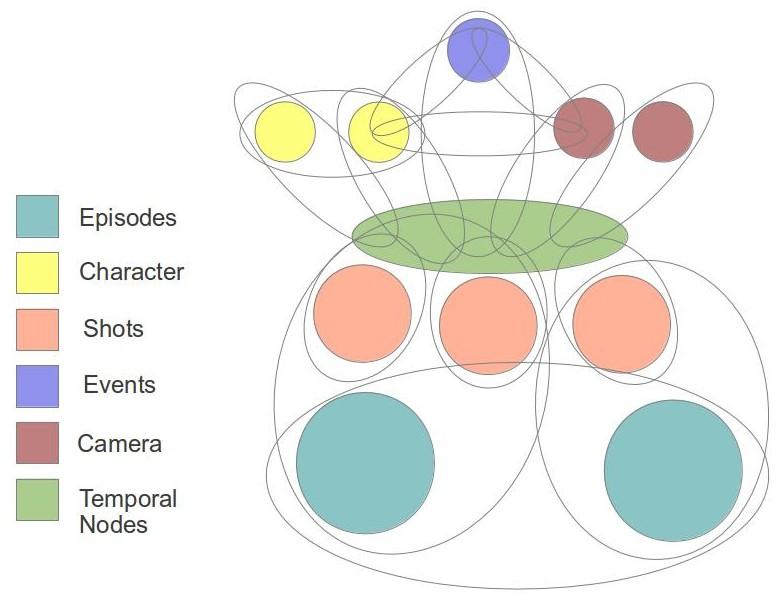
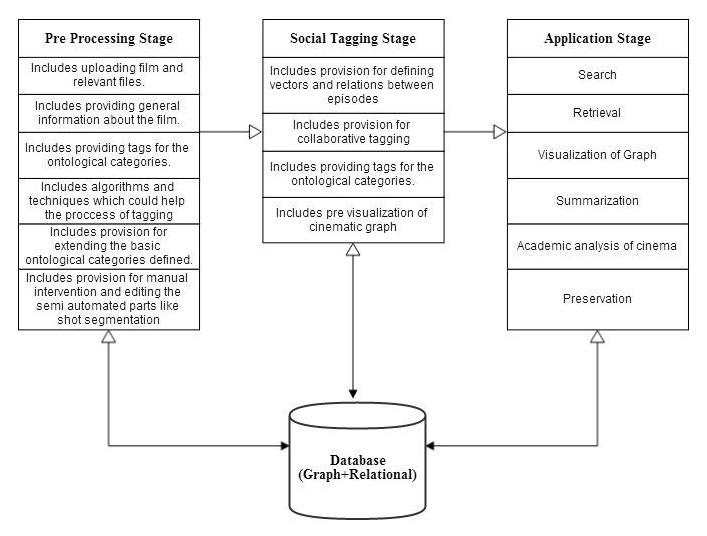


Figure 1. Venn diagram of cinematic hypergraph—data model.

# 

# **Architecture of CinemaScope**

The architecture of the tagging system we propose for CinemaScope is a web-based collaborative one. The annotation schema for the hypergraph data model in theory captures the whole graph. But in practice manual tagging, which is required for the annotating the themes, considerations, meanings, and vectors of cinema, reveals only a sub graph of the original graph. This is because the graph also contains codes and conventions (which are many times dependent on the individual viewer’s perspective) that regulate the narratives and enrich their pure meaning. It is therefore essential that the tagging of the cinema be made social so that most of the graph is captured. The high-level architecture of CinemaScope is given in Figure 2.

.

# 

# Figure 2. Basic architecture of CinemaScope.

# **Methodology of CinemaScope**

The database for the system is built on both relational database and graph database (HyperGraphDB). The relational database is used to store all the direct information about the cinema, like its cast, basic plot structure, related files (like script and subtitles), and other relevant information. The graph database is used to graph the tags which have a graphical relation between them. HypergraphDB is a graph-based as well as an object-oriented database with a Java-based API that allows objective modelling of all categories of tags. For example, ontological categories are modelled as follows:

public class OntologicalEntity{

private String entityName; //Entity Name

private Map<String, Object>attributesNameAndType = new HashMap<String, Object>(); //Contains the attributes with their names and values

public void setEntityName(String entityName){

//Setting The entity Name

}

public String getEntityName(){

return entityName;

}

public Object getAttributeValue(String attribute){

return attributesNameAndType.get(attribute);

}

public void fillAttribute(String value){

/\*Code for filling the attributes\*/

}

}

The pre-processing stage involves identification of shots, episodes, and the tagging of basic ontological entities. The pre-processing step cannot be completely automated owing to the limitations of current vision and audio processing algorithms. But this stage of the annotation can be aided with a number of supporting files and techniques. For example, movie screenplays and subtitles that are freely available on the Internet help in giving hints of annotation to the user. Even the processes that are completely automated require manual intervention and editing. There are various shot transition detection methods (Boreczky and Rowe, 1996) along with techniques for identification of camera properties like depth, movement, and angle (Benini et al., 2010). The camera properties identified are dependent on the definitions used in the methods, and they are not completely accurate. The tagging system should provide options for manually editing and supervising the shots and camera properties.

Film scripts and subtitles aid in the tagging of ontological entities by supplying hints. The script is first aligned with the subtitles for time stamps based on the work of Ronfard and Theung (2003). A basic version of this has been implemented based on the work of Larissa Munishkina et al. (2013). An example hint given by the system for the film *Indiana Jones and the Raiders of the Lost Ark* is as follows:

E.g. Scene #1 (0:00–2:15)

Characters: Indiana Jones, Baranaca

Locale: High Jungle, Peru

Things: Gun, Idol

Events: Shooting, Running

The Collaborative or Social tagging follows the pre-processing step, and it helps in the creation of tags and links in cinema. The user is given the provision of defining the relations, which is made possible by the use of HyperGraphDB. Hertzum et al. (2002) suggests that collaboration of film archives should facilitate different archives in identifying a common ground on which to base a collaborator and in acknowledging the distinctiveness of each archive. The architecture of CinemaScope allows identifying a common ground, in terms of pre-processing tags as well as distinctiveness, by allowing the users of different background to provide interpretations and considerations of the story in terms of vectors, relations, and links (each with a different label).

The various relations added by the user and the tags obtained from the pre-processing stage build up the graph, which could later be used for various applications like searching. For example, the following code snippet answers semantic queries like ‘Find all the camera properties used when character is holding the whip’:

HyperGraph graph = new HyperGraph(HyperGraphLocation);

List<CameraTemporalRelations> cameras = hg.getAll(hg.type(CameraTemporalRelations));

List<CharacterThingTemporalRelations> CTR = hg.getAll(hg.and(hg.type(CharacterThingTemporalRelations.class), hg.eq("spatial\_relation", "hold"), hg.eq("thing\_name","whip")));

# **Summary**

The paper describes the discrete theory of cinema and explains the hypergraph data model of cinema. The paper also describes CinemaScope, a semi-automated web-based collaborative film archive based on the hypergraph data model for annotating the film. This project is different from other movie annotation projects (ANSWER, 2009; Jewell et al., 2005); Lombardo and Damiano, 2010), which fail to capture the flow of cinema and make the computational representation of cinema in a database very descriptive and computationally infeasible.

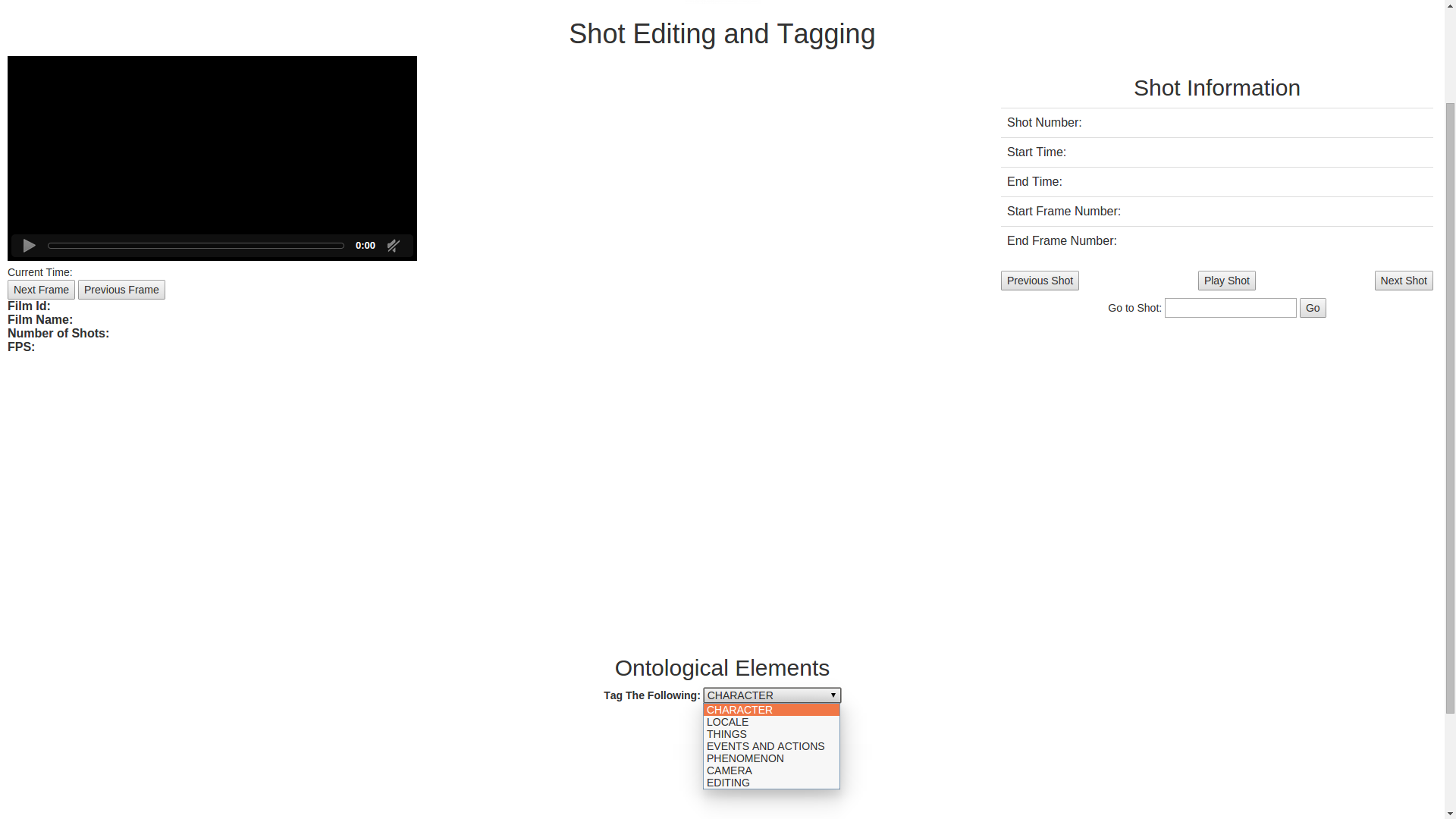


Figure 3. Prototype of CinemaScope annotation system.

# **References**

**ANSWER Annual Report.** (2009). http://cordis.europa.eu/fp7/ict/content-knowledge/docs/answer-annual-report-2009.pdf.

**Benini, S., Canini, L. and Leonardi, R.** (2010). Estimating Cinematographic Scene Depth in Movie Shots. Multimedia and Expo (ICME), 2010 IEEE International Conference.

**Boreczky, J. S. and Rowe, L. A.** (1996). Comparison of Video Shot Boundary Detection. *Journal of Electronic Imaging,* **5**(2) (April): 122–28.

**Chatman, S.** (1978). *Story and Discourse: Narrative Structure in Fiction and Film*. Cornell University Press, Ithaca, NY.

**Dhananjaya.** (1100). Dasarupaka, Dr. Keshavrao Musalgaunkar. Hindi Dasarupaka, 2004.

**Hertzum, M., Pejtersen, A. M., Cleal, B. and Albrechtsen, H.** (2002). An Analysis of Collaboration in Three Film Archives: A Case for Collaboratories. *CoLIS4: Proceedings of the Fourth International Conference on Conceptions of Library and Information Science.* Libraries Unlimited, pp. 69–83.

**Hochin, T.** (2006). Graph-Based Data Model for the Content Representation of Multimedia Data. Knowledge-Based Intelligent Information and Engineering Systems. *Lecture Notes in Computer Science,* **4252** (1182–90).

**Hochin, T. and Tatsuo T.** (2000). A Directed Recursive Hypergraph Data Model for Representing the Contents of Multimedia Data. *Memoirs of the Faculty of Engineering, Fukui University,* **48**: 343–60.

**Jewell, M., Lawrence, K., Tuffield, M., Prugel-Bennett, A., Millard, D., Nixon, M. and Shadbolt, N.** (2005). OntoMedia: An Ontology for the Representation of Heterogeneous Media. *Proceedings of SIGIR Workshop on Multimedia Information Retrieval.* ACM.

**Leibniz, G. W.** (2013). *The Labyrinth of the Continuum: Writings on the Continuum Problem 1672–1686*. Translated, edited, and with an introduction by Richard T. W. Arthur. Yale University Press, New Haven, CT.

**Lombardo, V. and Damiano, R.** (2010). Narrative Annotation and Editing of Video. Interactive Storytelling. *Lecture Notes in Computer Science,* **6432**, 62–73.

**Muni, B.** (1996 [300 BC]). *Nāṭya śāstra*. With Abhinavagupta’s Commentary; with Hindi Commentary by Dwivedi Parasnath. Sampurnanand Sanskrit Mahavidyalaya, Varanasi.

**Munishkina, L., Parrish, J. and Walker, M. A.** (2013). Fully Automatic Interactive Story Design from Film Scripts. Interactive Storytelling. *Lecture Notes in Computer Science,* **8230**: 229–32.

**Radev, I., Pissinou, N. and Makki, K.** (1999a). Film Video Modeling. *Proceedings of IEEE Workshop on Knowledge and Data Engineering Exchange, KDEX 99*, Chicago, IL, November 1999.

**Radev, I., Pissinou, N., Makki, K. and Park, E. K.** (1999b). Graph-Based Object-Oriented Approach for Structural and Behavioral Representation of Multimedia Data. *Proceedings of the Eighth International Conference on Information and Knowledge Management*, Kansas City, MO, 2–6 November 1999, pp. 522–30.

**Ronfard, R. and Thuong, T. T.** (2003). A Framework for Aligning and Indexing Movies with Their Script. *Proceedings of ICME*, Baltimore, MD, 6–9 July 2003.