Modeling Data of Immersive Environments*

Cyrus Shahabi, Mehdi Sharifzadeh and Albert A. Rizzo
Integrated Media Systems Center & Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781
[shahabi,sharifza,arizzo]@usc.edu

ABSTRACT

We propose a model to conceptualize various data sets pertaining to immersive environments. We argue that it is absolutely critical to understand the data generated by and for immersive environments for several purposes. The first step towards this understanding is to be able to model these data and then use the model for query and analysis of the data. Through several examples from current prototypic immersive applications we demonstrate the usefulness, effectiveness and generality of our proposed model.

Categories and Subject Descriptors

H.2.1 [**Information Systems**]: Database Management—*Logical Design*

General Terms

Design

Keywords

Immersidata, Immersive environment, Data modelling

1. INTRODUCTION

Immersive Environments allow a user to become immersed within an augmented or virtual reality environment in order to interact with people, objects, places, and databases. Many prototypic immersive environment applications already exist [1, 4, 5, 6, 7, 8]. While information on human performance in these environments can be captured in databases, analysis of this rich source of information is often neglected in many of the current prototypes. More effective capture and analysis of such immersive data could serve to inform efforts to improve system performance as well as enhance usability by guiding the development of more natural and customizable user interfaces.

*This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and IIS-0082826, and unrestricted cash gifts from Microsoft, NCR, and the Okawa Foundation.

ITP2002 Juan Les Pins, France

In the current paper, we provide a framework for the modeling of immersive data based on several prototypic immersive environments [5, 7, 8] including immersive applications developed at the USC's Integrated Media Systems Center (IMSC) [4, 1] and Institute of Creative Technology (ICT) [6]. The first step towards managing immersive data is to understand the various types of data that are characteristics of immersive environments. A data model is then needed to conceptualize these data sets. Finally, the effectiveness of the model should be studied by investigating the range of alternative query and data analysis methods that can be formalized using the model.

Based on some general definitions and categorization of immersive environments from a data management perspective, we identify three main data sets of immersive environments: 1) immersive space data, 2) user data and 3) interaction data. We show that the first set can be conceptualized as spatio-temporal data and has some commonalities with video data sets and their modeling. The second set is similar to traditional record-based data and includes user responses to an offline questionnaire about his/her experience. The third group is the data acquired from a user's interactions with an immersive environment, termed *Immersidata* [2]. In order to facilitate a natural interaction (beyond keyboard and mouse), the users in typical immersive environments are traced and monitored using tracking devices on their heads, hands, and legs, video cameras and haptic devices.

In a previous paper [2], we discussed the implementation issues for storage and access of immersidata. In this paper, we focus our attention on the modeling aspects of immersidata. In this regard, we propose a model that conceptualizes all three data sets relevant to immersive environments. We show the effectiveness of our model by using it in the formalization of a wide range of complex queries from knowledge discovery to spatio-temporal queries. To be specific, we first describe a particular immersive application as our running example. However, when appropriate, we refer to other prototypic applications to illustrate the generality of our model.

2. IMMERSIVE CLASSROOM APPLICATION

The *Immersive Classroom* (IC) is a Head Mounted Display (HMD) immersive virtual environment designed to test attention performance in normal children and those diagnosed with Attention Deficit Hyperactivity Disorder (ADHD) [1]. The objective of the application is to differentiate these two user groups by analyzing their performance within the IC. The IC virtual environment consists of a typical classroom with student and teacher desks, a virtual teacher, a black-

board, a large window looking out onto a playground with buildings, vehicles, and people, and a pair of doorways on each end of the wall opposite the window through which activity occurs. A typical IC task consists of alphabetic characters that are sequentially displayed on the blackboard with the child instructed to press a button when a particular letter pattern is seen. For example, the "AX attention task" requires users to press a mouse button as quickly as possible upon detecting an X preceded by an A (a hit) and to withhold responses to any other pattern. At the same time, a series of typical classroom distractions are systematically manipulated within the IC (i.e., ambient classroom noise, paper airplane flying around the room, students walking into the room, activity occurring outside the window). In addition to the attention response measures, trackers placed on the head, hands and legs monitor hyperactive body movements of the child and stream the data continuously. Upon collecting such immersidata over a series of trials, psychologists would like to be able to ask a variety of queries about the stored data set. The queries can be as simple as: "Which distraction was present when a particular child missed a target?", to more complex queries such as "Automatically distinguish ADHD children from normal ones."

3. DEFINITIONS

The main components of an immersive environment are user (*subject(s)*), *virtual space*, *actor objects* and a task objective, *mission*. We define each of these in turn.

Subject is a human user who performs a task in the environment. During the task performance, the subject interacts with virtual objects and/or other subjects in the environment. Three types of data are associated with each subject:

1. *Profile* that includes subject personal information such as her age, gender, abilities or even medical history.
2. *Preferences* that is the way she prefers the environment to look and feel.
3. *History* that is the *state* [9] of the subject and may include performance information from previous experiments or when the previous experiment interrupted.

Virtual space is a simulation of real world. This space can be generated in different ways. Virtual Reality (VR) techniques can be used to render 3-D graphics (like a typical classroom in the IC application) where a subject wears a HMD to observe the environment. In other applications such as Office of the Future [7], the space can be a single *projected video* of a real office on a white wall. Alternatively, projected objects can be displayed on parts of a room such as walls or tables using data projectors (e.g., the Everywhere Displays project [5]). In all three cases the space is defined as geometry of some *scene objects*. These are static objects like chairs, desks and blackboard in the IC application or a selection menu in [5].

Actor objects are objects whose appearance or disappearance in the space or their changes in position during the system lifetime are main causes of system events. Actor Objects are typically classified based on their presence types: *visual, audio, odorous* or combinations of these three. More complex events are recognized from combining basic events to form actor objects' behavior.

Mission is the task which subject is asked to perform while immersed in the environment. The tasks can be as simple as pressing a button after observing an actor object or as complex as rescuing a family from a fire in their house and putting the fire out (e.g., FireFighter Training System [8]). In the very simple case such as with IC, subject has to interact with the system following a sequence of events which matches a predefined pattern.

4. MODELING IMMERSIVE DATA

In this section, we conceptualize immersive environments using an object-relational data model. While this model may not be able to cover the aspects of all immersive environments but it is a typical framework for formally representing common components of such environments.

4.1 Entities

The entities describe different data sets generated by and maintained for an immersive environment. To simplify the discussion, we classify the entities into three groups. The first group of entities conceptualizes various traditional data sets within an immersive environments. The conventional relational model would suffice to describe these data sets. The second group describes the spatial and temporal data sets. The object-relational model is used to describe these data types. For implementation purposes, an object-relational database such as Informix or Oracle 9i with spatial and temporal extensions (datablades or cartridges) can be used. The modeling of this group is very similar to work on modeling video data [10, 11, 12] except in 3-D. The other main difference is that spatial, geometry and temporal aspects of objects do not need to be extracted using image recognition techniques, but provided accurately by the rendering subsystem. The last group of entities conceptualizes data generated as a result of subject(s) interactions with the immersive environment, termed *immersidata*.

Throughout this section, the *Position* attribute in some of the entities refers to the *Minimum Bounding Box (MBB)* of the object in 3-D space. The position is determined based on a reference coordinate space. Now consider each group in turn.

4.1.1 Conventional Data

- $Subjects(SID, PData, HData, Pref)$

This entity is a representative of the subject(s) in the environment. SID is a unique identifier of each subject. $PData$ is a composite attribute including subject personal information such as age or health status. $HData$ is a reference to another entity (i.e., foreign key) conceptualizing the data on how the subject has performed the designated tasks during previous experiments. $Pref$ refers to subject preferences about the space.

- $SubjectInput(SID, T, Response)$

This entity includes the subject answers to the system using traditional input devices (e.g., mouse clicks for menu selection or keyboard strokes for text responses). SID is the subject identifier, T is the time of answering a task and $Response$ is the value of the answer as a character string.

- $OfflineInput(SID, QID, Response)$

This entity specifies the subject SID's answer to the question QID that is part of an off-line questionnaire about the mission. $Response$ is the value of the answer.

4.1.2 Spatio-Temporal Data

- $SceneObjects(SOID, Name, Position)$

This entity contains geometry of scene objects that constitute the virtual space. $SOID$ is a unique identifier for the object. $Name$ is an attribute to keep object's description. The object is not movable and $Position$ is its 3-D placement in the reference coordinate space.

• $ActorObjects(AOID, Name, Type)$

Actor objects are dynamic moving objects in the environment. $AOID$ is a unique identifier. $Name$ is a descriptive attribute. $Type$ keeps object's presence type such as *visual, audio,* and *audio-visual*.

• $ActorObjectsTrajectory(AOID, TP_1, TP_2, ..., TP_n)$

The entity represents Actor Object movements, where TP_i is the tuple $(Time, Position)$. A special value for position can be specified to represent object disappearance. Non-visual objects are specified by the position of their source (e.g., speaker).

• $Events(EvID, T_1, T_2, P, O_1, O_2, ..., O_n)$

We describe the application specific events [9] using this entity. $EvID$ is a serial incremental identifier of the event that has occurred from time T_1 to T_2 at position P, in which all objects or subjects O_i have been involved. If O_i refers to a subject, the event entity can be classified under the third group of entities, i.e., immersidata.

4.1.3 Immersidata

• $BodyPositions(SID, PartId, T, Position, Rotation)$

This entity specifies the position of the part $PartId$ of the subject SID at time T as well as the *Rotation* of the part in the reference XYZ coordinate space. In a *sensing system*, $PartId$ is associated with a particular tracker/sensor.

• $SubjectTrajectory(SID, TP_1, TP_2, ..., TP_n)$

This entity includes subject locations in certain times. Each TP_i is a $(Time, Position)$ pair. This can be a derived entity (or *view*) by applying an *average* function on entity $BodyPositions$ over different body parts of subject SID.

• $SubjectSight(SID, T, Sight)$

This entity captures the area that subject SID can see at a given time T (i.e., line of sight). A common approach to extract *sight* is using a *camera model*. The camera model assumes that the subject is looking towards the center of the display and uses the bounding area made by the planes passing through the display edges and the eye point. Thus, *sight* consists of the set of these planes.

• $Gestures(SID, N, T_1, T_2, Sign)$

This complex entity contains detected signals from subject body movements. It is the result of an aggregation on $BodyPositions$ over position and rotation data, and applying a similarity measure to recognize a specific command or sign from a known library of commands or signs (e.g., American Sign Language). Subject SID has performed some movements from time T_1 to T_2 to input her Nth sign and $Sign$ is an identifier for the recognized command[1].

4.1.4 Primitive Functions

For the remainder of this paper we use tuple relational calculus, a variant of relational calculus, to describe different types of function primitives and queries [13].

Two abstract primitive functions should be discussed in order to describe future queries:

[1]The detailed description of pattern/sign recognition from immersidata is beyond the scope of this paper and is discussed in a separate paper [3].

• Euclidian distance function which computes distance between object(s) and/or subject(s) and/or a specific location at time T based on a reference coordinate system: $Distance(SID|AOID|SOID, SID|AOID|SOID|P, T)$. This is provided as a basic operation in spatial databases.

• $IsInSight(SID_1, SID_2|AOID|SOID|P, T)$, which specifies whether subject SID_1 can see another subject SID_2, actor object $AOID$, scene object $SOID$, or location P at time T. This function can be implemented using a camera model as described in Section 4.1.3.

4.1.5 Applied Examples

With the IC application, we can define presence of a character (an actor object) on the blackboard and the AX event, respectively, as:

- $Characters(T, N, C)$ represents showing character C which is the Nth character in the stream at time T.
- $AXEvents = \{C_{12} \mid \exists\, C_1, C_2 \in Characters\ (\ (C_2.N = C_1.N + 1) \wedge (C_1.C = A) \wedge (C_2.C = X) \wedge (C_{12}.T_1 = C_1.T) \wedge (C_{12}.T_2 = C_2.T) \wedge (C_{12}.N = C_1.N)\ )\}$ is the AX pattern on the characters and the time that they appeared on the blackboard. This can be implemented by applying a semi-join on $Characters$ that finds subsequent characters X over all characters A.

An alternative event is $DoorOpened(AOID, T, D, P)$ which specifies the event of opening the door $AOID$ at location P from time T to $T + D$.

5. QUERYING IMMERSIVE ENVIRONMENTS

Our proposed model can be used to support many interesting queries on an immersive environment. To illustrate, we start by using our model to formalize subject *attention* as an example of a new data mining measure for knowledge discovery from user interaction data (immersidata). Subsequently, we discuss some more typical queries that can be formalized using our model.

5.1 Subject Attention Query

Subject attention is a common behavioral concept that can be identified by analyzing user interactions. This concept is useful to evaluate the subject performance during or after a mission or to modify the environment by eliminating those distractions that are shown to negatively impact attention. Moreover, by detecting a distraction in a feedback driven system, the system can modify the position of actor objects in real-time in order to bring the subject's attention back to the task.

We define subject attention based on how much a user has been distracted while performing a task. If the environment is conceptualized using our proposed model, appearance of a visual actor object or an event happening within a certain distance of subject can potentially be a distraction. If the user's head moves in the direction of the object or sound, a distraction has occurred. We limit our search to the distance threshold D_t from subject:

$Distractions = \{D \mid \exists\, t, \exists\, S \in Subjects, \exists\, E \in Events$
$(\ (E.T_1 \le t \le E.T_2) \wedge (Distance(S.SID,\ E.P,\ t) < D_t) \wedge$
$(IsInSight(S.SID,\ E.P,\ t) = True) \wedge (D.SID = S.SID) \wedge$
$(D.EvID = E.EvID) \wedge (D.T_1 = E.T_1) \wedge (D.T_2 = E.T_2)\ )\}$

This can be implemented by first applying a relational join operator on entities $Subjects$ and $Events$. Next, we need to

filter (i.e., selection operator) those objects in the resulting events that are both close to the subject and within her line of sight. Rewriting the above query using *DoorOpened* event in IC we can find those distractions attributed to the opening of a door. This query searches for the times t in which subject SID has been looking at the door $AOID$ that is closer to her than the distance threshold D_t:

$Distractions = \{D \mid \exists\, t, \exists S \in Subjects, \exists\, DO \in DoorOpened$
$(\ (DO.T \leq t \leq DO.T + D) \wedge$
$(Distance(S.SID,\ DO.AOID,\ t) < D_t) \wedge$
$(IsInSight(S.SID,\ DO.AOID,\ t) = True) \wedge$
$(D.SID = S.SID) \wedge (D.AOID = DO.AOID) \wedge$
$(D.T = D).T)\ )\}$

5.2 More Query Types

Using our model a rich set of queries can be formalized to extract a subject behavior or search for patterns of interest while the subject is interacting with the environment. Below are some examples:

- **Spatio-temporal range queries** can retrieve components based on their both spatial and temporal dimensions. For example, *"Which events have been occurred within a distance D_t of the subject during the time that she has raised her hand?"*. To formalize this query, we first find the time interval that the subject S_{id} has her hand raised. Next, we find all the events happening during the same time interval. Finally, we filter those events using the given spatial range (D_t).

$Q_1(S_{id}) = \{E_1 \mid \exists S \in Subjects, \exists E \in Events,$
$\exists G \in Gestures\ (\ (G.SID = S.SID) \wedge$
$(G.Sign = RAISEHAND) \wedge (E.T_1 \geq G.T_1) \wedge$
$(E.T_2 \leq G.T_2) \wedge (Distance(S.SID, E.P) < D_t) \wedge$
$(S.SID = S_{id}) \wedge (E_1 = E)\ )\}$

- **Spatial/Temporal kNN queries**[2] are specified with a temporal range and a kNN predicate on spatial dimensions or vice versa. For example, *"Identify the closest five victims to the subject in the first 10 minutes of the mission in the FireFighter system"*. We use $NClosest(SID, T, N)$ as a primitive function provided by spatio-temporal databases for retrieving N nearest objects to the subject SID at time T.

$Q_2(S_{id}) = \{A_1 \mid \exists S \in Subjects, \exists A \in ActorObjects, \forall t \in [0, 10]\ (\ (A.AOID \in NClosest(S.SID, t, 5)) \wedge (S.SID = S_{id}) \wedge (A_1 = A)\ )\}$

- **Spatial aggregate queries** deal with abstractions of subject and/or object locations in space. For example, *"What is the average distance between the subject and all the window objects in the FireFighter system?"*

$Q_3(S_{id}) = \{AVG(d) \mid \exists S \in Subjects, \exists O \in SceneObjects,$
$\forall t\ (\ (O.Name = "window")\ \wedge\ (S.SID = S_{id}) \wedge$
$(d = Distance(S.SID, O.SOID, t))\ )\}$

- **Temporal aggregate queries** are specified by aggregating the temporal dimension of entities. For example, *"What is the average time it takes for a subject to give an arm signal during a traffic training system?"*

$Q_4(S_{id}) = \{AVG(t) \mid \exists S \in Subjects, \exists G \in Gestures$
$(\ (G.SID = S.SID) \wedge (G.Sign = ARMSIGNAL) \wedge$
$(t = G.T_2 - G.T_1) \wedge (S.SID = S_{id})\ )\}$

$G.Sign$ is detected by applying a pattern matching technique on subject's hand movements and is compared to a known sequence of movements as $ARMSIGNAL$.

[2]k-nearest neighbor search: finding k closest objects to a given location.

6. CONCLUSION AND FUTURE WORK

We proposed a general model for conceptualizing immersive data. Through several empirical examples we demonstrated that our model can formalize complex data mining and spatio-temporal queries that are critical for understanding users' behaviors in immersive environments. We intend to implement our model using an object-relational database and populate the database with data collected from an immersive application and the real-world interactions with the application. Finally, we will use the implemented system to evaluate the effectiveness of the model and perhaps for extending the model.

7. REFERENCES

[1] A.A. Rizzo, et. al, Virtual Environments for the Assessment of Attention and Memory Processes: The Virtual Classroom and Office, Proceedings of The 4th International Conference on Disability, Virtual Reality and Associated Technology, Veszprm, Hungary, 2002.

[2] C. Shahabi, AIMS: An Immersidata Management System, CIDR'03: Conference on Innovative Data Systems Research, Pacific Grove (CA), January 2003.

[3] C. Shahabi and D. Yan, Sequence Matching and Separation of Immersive Sensor Data Streams, Submitted for Publication, 2002.

[4] M. McLaughlin, et al., The Haptic Museum, Proceedings EVA 2000 Conference on Electronic Imaging and the Visual Arts, Florence, Italy, 2000.

[5] R. Kjeldsen, et al., Interacting with Steerable Projected Displays, Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition (FG'02), Washington (DC), May 2002.

[6] W. Swartout, et al., Toward the Holodeck: Integrating Graphics, Sound, Character and Story, in Proceedings of 5th International Conference on Autonomous Agents, Montreal, Canada, June 2001.

[7] G. Welch, et al., Projected Imagery In Your Office in the Future, IEEE Computer Graphics and Applications, pp. 62-67, July/August 2000.

[8] Training Virtual Environment Project, Virtual Environments Group, Georgia Tech, http://www.cc.gatech.edu/gvu/virtual.

[9] R. Jain and A. Katkere, Experiential Environment for Accessing Multimedia Information, Unpublished Draft.

[10] W. Al-Khatib and A. Ghafoor, An Approach for Video Meta-data Modeling and Query Processing, Proceedings of ACM Multimedia, pp. 215-224, 1999.

[11] S.-C. Chen and R.L. Kashyap, A Spatiotemporal Semantic Model for Multimedia Presentations and Multimedia Database Systems, IEEE Transactions on Knowledge and Data Engineering, vol. 13, no 4., pp. 607-622, 2001.

[12] L. Chen and M. T. Özsu, Modeling Video Objects in Video Databases, Proceedings of IEEE International Conference on Multimedia and Expo(ICME), Lausanne, Switzerland, August 2002.

[13] R. Ramakrishnan and J. Gehrke, Database Management Systems, second edition, Mc Graw Hill, 2000.