MED: A Multimedia Event Database for 3D Crime Scene Representation and Analysis

Marcin Kwietniewski¹, Stephanie Wilson¹, Anna Topol¹, Sunbir Gill¹, Jarek Gryz¹, Michael Jenkin¹, Piotr Jasiobedzki² and Ho-Kong Ng²

¹York University, Department of Computer Science and Engineering
⁴7000 Keele St., Toronto, ON, Canada, M3J 1P3
²MDA, 9445 Airport Rd., Brampton, ON, Canada, L6S 4J3

Abstract—The development of sensors capable of obtaining 3D scans of crime scenes is revolutionizing the ways in which crime scenes can be analyzed and at the same time is driving the need for the development of sophisticated tools to represent and store this data. Here we describe the design of a multimedia database suitable for representing and reasoning about crime scene data. The representation is grounded in the physical environment that makes up the crime scene and provides mechanisms for representing both traditional (forms-based) data as well as 3D scan and other complex spatial data.

I. INTRODUCTION

Crime scene investigation requires the collection and analysis of vast amounts of data. This data comes in a variety of formats, such as traditional paper forms collected by officers at the scene, results of scientific evaluations from the crime lab, fingerprint data, high resolution still imagery, and video and audio recordings. In a typical scene hours of video data, and thousands of images and other separate pieces of evidence may be collected. Searching and referencing this data is a tedious and labour intensive task.

Representing this information in an effective manner in order to enhance later investigation of the crime is a complex task. This task becomes even more difficult as advanced technologies such as laser [4] or optical scanners [7], [6] become available which allow for full 3D scans of the scene and objects within the scene to be collected and made available for investigators. It is important for investigators to be able to access this large and varied collection of data in an intuitive and efficient fashion.

Crime scene data poses several challenges for its storage and processing by a database system. In particular, they pose challenges related to:

- **The size of the data.**
  The iSM sensor [7] collects 500GB of data per hour and it may take an hour to acquire a large crime scene. The 3D model rendered from the raw data is an order of magnitude larger; a model of a single crime scene can require terabytes of storage.

- **The complex relationships between database objects.**
  Many of the objects represented in the database are related through a hierarchical order, but even more complex relationships are present. In general, entities are associated with larger structures: They may contain one another (for example, the gun is in the drawer) or the process of data collection enforces specific relationships between entities (for example, the coroner’s report is associated with the body of the victim). Another property of crime scene data, especially the data associated with 3D scene acquisition and other sensor systems, is that the data may be available at a variety of different resolutions. The use of different resolutions may be related to performance issues (for example, it is more efficient to render the 3D representation of the crime scene with fewer polygons), or it may be necessary to view the crime scene with certain objects removed or obscured (for example, to protect the identity of the victim).

- **The large variety and inefficiency of functions and tools for interacting with 3D objects.**
  A typical database system provides few distinct ways of representing the same piece of data. For 3D data, there is a wide spectrum of such representations. A user may retrieve an object from a database and modify it in various ways (for example, by taking a 2D snapshot, changing resolution, removing certain objects from it, etc.). These operations may be time consuming and it may be necessary to store these user-generated objects (views) for future use.

- **The importance of the user interface.**
  The client interface must provide the user with the ability to easily interact with and modify the data. From the database system perspective, the hardest task is to clearly describe the relationships between objects (for example, that one object is a 2D snapshot of the scene or a low resolution version of it). There is no formal language like SQL to describe precisely to the user how the objects are related to each other. As the number of such objects becomes large, the task of representing efficiently and clearly their relationships becomes a challenge.

This paper describes the design of a multimedia database to address the problem of representing and manipulating the complex datasets associated with crime scene analysis. The user interacts with the database through a user interface that provides them with a 3D view of the crime scene. The user
can ‘walk’ through the resulting 3D model querying objects that are encountered (see Figure 1). Interactions – either traditional graphical operations or more complex queries of the database – are processed by the underlying database engine that determines what should be displayed as well as retrieving the set of actions that can be performed on the displayed objects.

3D modeling systems such as instant Scene Modeler (iSM) [7] and CBRN Scene Modeler (C2SM) [2], both developed by MDA and used in this project, collect and process images to create photorealistic models. For example, iSM was used to collect approximately 3,000 stereo images (20 GB) during two hours of data acquisition in a simulated crime scene that consisted of three rooms. The resulting 3D model is shown in Figure 1. After processing, the 3D model size (3D geometry and colour texture) is 50-200 MB (depending on the texture resolution). Most of the visualization and measurements can be performed using 3D models alone. However, for the purpose of investigation and potential court proceedings it is crucial to retain all the original data, and to provide access to it. The new system, C2SM, records also high resolution digital still images (10M pixel) from an additional camera; with this option the system, C2SM, records also high resolution digital still images to create photorealistic models. For example, iSM was developed by MDA and used in this project, collect and process images to create photorealistic models. For example, iSM was used to collect approximately 3,000 stereo images (20 GB) during two hours of data acquisition in a simulated crime scene that consisted of three rooms. The resulting 3D model is shown in Figure 1. After processing, the 3D model size (3D geometry and colour texture) is 50-200 MB (depending on the texture resolution). Most of the visualization and measurements can be performed using 3D models alone. However, for the purpose of investigation and potential court proceedings it is crucial to retain all the original data, and to provide access to it. The new system, C2SM, records also high resolution digital still images (10M pixel) from an additional camera; with this option the size of raw data would have been 50 GB. In general, the amount of storage space required for a specific scene depends on its size, complexity, required resolution and coverage, and operating mode. It can easily reach (raw data and processed models) hundreds of GBs.

II. DATABASE DESIGN

A database for crime scene investigation carries dual functionality. It provides persistent storage of the raw data acquired from the crime scene and a method of analysis of the scene data for crime scene investigators. The first function is important for providing evidence in a court of law. Here, the data must be represented in an unaltered state so that the court can view the evidence as it was when it was recorded. The second function is important to aid the investigators by providing the data they require in a quick and easy to understand manner. It is important to be able to find/identify objects of interest in the scene. The representation must also provide straightforward access to the spatial and other relationships that exist between entities. For example, it is important to be able to identify entities that are in close proximity to each other and to represent the property that certain entities belong to others. This spatial hierarchical property of the data suggests a hierarchical representation for the database itself. At the top level there is the overall crime scene. The children of the root-level may include entities such as a chair or a table in the room. The children of the table may include a box resting on top of it. The children of the box may include a pencil inside the box, and so on. In order to simplify establishing relationships between objects, each object is assigned a bounding volume and children of a node must exist within this volume. Two children of a given node may have intersecting bounding volumes.

In addition to providing a mechanism for representing the property that some objects are contained within others, we need a way to reflect the fact that certain objects in a crime scene may have significant information associated with them that was collected at different sites (for example, at the crime lab or in coroner’s lab) or at a different time. Those representations or descriptions can be treated as separate ‘scenes’ in terms of processing and each node in the hierarchy is associated with a set of them. This provides a straightforward mechanism to generate views of the scene in the browser application, for example, it is able to query lower quality representations of a subset of the object hierarchy that it is necessary to render.

Objects within the hierarchy may have a range of different primitive type representations associated with them, including

- ‘Traditional’ descriptions like standard paper forms.
- Multimedia objects such as video or audio clips that describe (or encode) an object positioned at a specific location.
- Textured 3D meshes that describe the surfaces of objects and locations within the crime scene that can be easily added or removed from a 3D view of a crime scene.
- Spatial distributions such as levels of radiation, temperature, or gas concentrations.

Given the highly visual nature of the data that is encoded in a crime scene dataset, and the needs of crime scene investigators, the ability to visualize the dataset in various ways is a critical requirement. This raises a number of interesting problems related to both performance (at what resolution should large triangular meshes be rendered) and also to the security of views of the crime scene. It may be desirable, for example, to display portions of the crime scene in such a manner as to obscure certain facts that are being withheld during an ongoing investigation. Thus objects in the representation, either external or internal nodes in the hierarchy, may have a range of visualization attributes that
and clearly may be a challenge. If the number of such objects becomes large, the task of representing their relationships efficiently is also redundant objects in a database. The task of storing materialized views, which from a conceptual point of view are also redundant objects in a database. The task of storing these objects in our environment is easier in the sense that we do not expect updates to our original data, hence we do not need to solve the maintenance problem of the user-generated objects. On the other hand, we do not have a formal language like SQL to describe precisely to the user how the objects are related to each other. If the number of such objects becomes large, the task of representing their relationships efficiently and clearly may be a challenge.

III. DATA EXCHANGE AND USER INTERACTIONS

The client application and the database must be able to unobtrusively exchange information about the objects, their representations, their relationship with other child/parent objects, and available object manipulation actions. The Extensible Markup Language (XML) is well suited for sharing the information organized in a tree-based structure between systems and decouples the database interface and the user interface implementations. XML is used to describe queries and responses to the database as well as encoding the set of possible interactions the user can make with respect to a specific object in the database.

In order to ensure high-performance for rendering and interaction with a large number of 3D models, audiovisual, and other data the end-user client application is written natively utilizing operating-system supported hardware accelerated libraries (see Fig. 2). The task of decoupling interactions from the client application thus requires a mechanism at run-time to bridge the gap between the XML response from the database and the desired result in the client application. This is achieved using Python [5], a dynamic, scriptable, multiplatform, freely available object oriented programming language.

When an object is retrieved from the database the set of possible interactions and the Python code necessary to process these interactions is retrieved from the database as an XML description. For example, if the object is displayable at different resolutions, then the available commands to choose these resolutions are encoded in the XML description as is the set of actions that must be executed in order to cause the user interface to display the object.

When a user selects an object of interest in the scene available actions associated with that object (as identified in its XML description) are displayed. The manipulations may include viewing the object at a finer resolution, opening associated reports or audio files, or even removing the object from the scene altogether. Each such action will trigger the embedded Python script to interact with the database, which in turn will produce a new XML formatted description of the scene and the object according to the latest query (Fig. 3). The application will then use this information to return the results to the user. This may include redrawing the scene or launching a document viewer or a media player.

The XML response from the database to the client is an arbitrary length list of available interactions. These interactions are defined by two fields: a UI description tag and an inline Python script. The UI description tag is used by the client application to describe the action to the user. The corresponding Python script drives the client application.

This approach of separating the database interface and the user interface implementations provides greater potential for future extendibility. By using the hierarchical object association and the XML to return the query results we can communicate arbitrarily complicated scenes between the database and the application that avoids any direct mapping to the database tables.

IV. SEMANTIC SEMI-AUTOMATIC OBJECT SEGMENTATION

The 3D models of the crime scene environments obtained with a stereo sensor are not segmented in terms of underlying components of the scene. To be able to work with the semantic objects in the scene independently of the background scene itself it is necessary to segment these objects out. A semi-automatic tool is used allowing the user to indicate what object is to be segmented while the tool selects the specific area that corresponds to the desired object. The tool used is an extension of 2D lazy snapping [3] to 3D.

The 3D polygon or point cloud representation is converted into a discrete 3D voxel grid. The size of the voxels is variable, allowing the user to choose the ideal balance between speed of processing and accuracy of segmentation. The user then indicates which object is to be segmented by tagging foreground and background seed voxels, as shown in Figure 4 (c). This is done by simply painting with a brush the desired voxels to be tagged. An arbitrary number of voxels may be
tagged and the view of the 3D scene can be translated and rotated to tag voxels that may be hidden in some views.

The problem of segmenting the 3D dataset can then be expressed as a graph cut problem, with the cut dividing the foreground voxels from the background voxels. A graph is created, where a node in the graph is used to represent each non-empty voxel and adjacent non-empty voxels are considered connected nodes. A sink node, representing the background, and a source node, representing the foreground, are also added to the graph, with edges connecting them to each voxel node. Graph edge weights are established so that edges connecting similarly coloured voxels have a high weight, while edges connecting dissimilar voxels have a low weight. A penalty term is assigned to adjacent nodes along the segmentation boundary proportional to the difference in colour. The edge weight between these adjacent nodes thus increase in proportion with the colour similarity of the two nodes, so that similar colour nodes are less likely to be on different sides of the segmentation boundary. The graph cut is found using the min-cut/max-flow algorithm [1]. In the resulting cut, source connected nodes are foreground nodes while sink connected nodes are background nodes.

Figure 4 shows the process of 3D lazy snapping. The algorithm successfully segmented the body from the remainder of the image, based on the user’s initial seeding.

V. SUMMARY

The developed multimedia system combines diverse aspects of complex multimedia databases with an “immersive” visual display to support crime scene investigation. It bridges innovative technologies such as photo realistic 3d crime scene scans and traditional investigative documentation methods. The database requirements for representing non-homogeneous spatial datasets such as those required for crime scene investigation are novel and complex. Data must be represented at a variety of resolutions and must be grounded in the underlying spatial representation. It must also be possible to associate varied data in a hierarchical manner with objects within the scene. At the front end the user interaction with a “walkable” 3d interface is intuitive and easy to use. Underneath, the future extendibility is achieved through the use of a scripting language separating the user interface and database interaction implementation. This innovative system promises to enhance the efficiency of the investigation process.

VI. ACKNOWLEDGEMENT

This work was supported by the CBRN Research Technology Initiative (project number CRTI-05-0122) and MDA.

REFERENCES