

A Conceptual Model of Intelligent Multimedia Data Rendered using Flying Light Specks

Nima Yazdani
University of Southern California
Los Angeles, CA, USA
nimayazd@usc.edu

Hamed Alimohammadzadeh
University of Southern California
Los Angeles, CA, USA
halimoha@usc.edu

Shahram Ghandeharizadeh
University of Southern California
Los Angeles, CA, USA
shahram@usc.edu

ABSTRACT

A Flying Light Speck, FLS, is a miniature sized drone configured with light sources to illuminate 3D multimedia objects in a fixed volume, an FLS display. A swarm of FLSs may provide haptic interactions by exerting force back at a user's touch. This paper presents a conceptual model for the multimedia data to enable content-based queries. The model empowers users of an FLS display to annotate the illuminations by adding semantics to the data, extending a multimedia repository with information and knowledge. We present a core conceptual model and demonstrate its extensions for two diverse applications, authoring tools with entertainment and MRI scans with healthcare.

Holodecks Reference Format:

Nima Yazdani, Hamed Alimohammadzadeh, and Shahram Ghandeharizadeh. A Conceptual Model of Intelligent Multimedia Data Rendered using Flying Light Specks. *Holodecks*, 1(1): 38-44, 2023.
doi:10.61981/ZFSH2309

1 INTRODUCTION

There is an abundance of 3D data and devices that produce this data. They pertain to diverse applications ranging from healthcare to entertainment. For example, with healthcare, MRI scanners produce 3D scans of a patient's brain. With entertainment, a graphic artist may use a 3D authoring tool such as Maya or Blender to produce animated sequences. Flying Light Specks [3, 9, 10], FLSs, enable 3D displays to render this data without sacrificing either its metadata or the raw device data. A conceptual challenge is how these displays will integrate into the eco-system of today's devices to illuminate their 3D data and facilitate advanced interactions. This paper presents a conceptual data model to address this challenge. Its objective is to articulate the requirements at a high level to engage all stake holders, both technical and non-technical. The model will facilitate future logical (mathematical) and physical (software and hardware) models. The proposed data model is extensible, enabling a designer to extend it with new concepts specific to an application.

Figure 1 provides an end-to-end overview of a framework that includes the proposed conceptual data model. At one end of the spectrum, devices and authoring tools produce raw data with diverse metadata. This metadata may be rich in content. For example, the metadata authored by a graphic artist using a 3D authoring

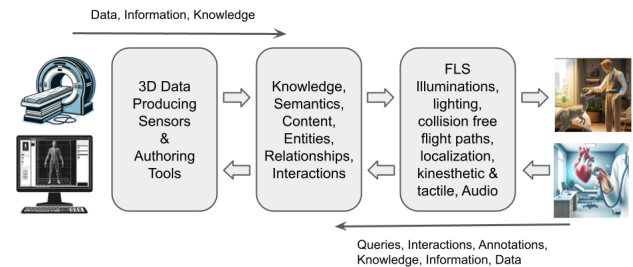


Figure 1: An eco-system for devices and authoring tools, and 3D FLS displays.

tool may pertain to settings of algorithms that perform complex tasks, e.g., keyframing for interpolations that produce animated sequences [14]. This data is abstracted and transformed into information, a summarization that is more interpretable by humans, e.g., MRI scan of the patient's brain shows dark matter. Finally, data and information is transformed into knowledge that provides an understanding of the content of the physical and virtual worlds, e.g., the patient has seizures because of head injury.

At the other end of the spectrum, a 3D FLS display provides its users with illuminations and diverse forms of interactions including haptics. Users may query the system for the relevant data, information, and knowledge. In addition, users may annotate the illuminations to contribute additional data, information, and knowledge to the system.

In this paper, we present a conceptual model that describes the end-to-end spectrum of Figure 1. The immediate benefit of the conceptual data model is two folds. First, it facilitates design and implementation of efficient algorithms for the FLS display. Second, it enables content-based queries and annotations that contribute knowledge, realizing intelligent multimedia data. Consider each in turn. An algorithm that enables a 3D FLS display may use the semantics of multimedia data. To illustrate, consider an animated sequence produced by an authoring tool. To illuminate this sequence, an FLS display must compute the flight path of FLSs as a function of time. This can be done in several ways. In [10], the animated sequence is reduced to a sequence of mesh files that are subsequently processed using the Motill [10] algorithm. While highly parallelizable, Motill is a resource intensive algorithm that detects motion across the points of different mesh files. A more efficient algorithm may use the proposed conceptual model that identifies

the key frames that have motion. This may be the basis of a new algorithms with a significantly lower complexity than Motill. Similarly, with content based queries, we may use the information and knowledge readily available from devices and authoring tools to provide relevant results. This will enable language models that understand and respond to domain specific queries.

The **contributions** of this paper include:

- (1) A core conceptual data model for intelligent multimedia data displayed using an FLS display. Section 3.
- (2) Extensions of the core to MRI healthcare data and authoring tools for animation. Sections 4 and 5.

Section 2 presents related work. Brief conclusions and future research directions are presented in Section 6.

2 RELATED WORK

The Flying Light Speck, FLS, as a miniature sized drone configured with light sources, processing, storage and networking capabilities to illuminate complex multimedia shapes was first introduced in [9]. 3D FLS displays [2, 9–11] are in the same class of systems that materialize virtual objects in a fixed volume. These include fast 3D printing [7], Claytronics as physical artifacts using programmable matter consisting of catoms [12], Roboxels as cellular robots that dynamically configure themselves into the desired shape and size [16], BitDrones [13] and GridDrones [4] as interactive nano-drones.

Motill [10] is an algorithm that computes the flight path of FLSs for an animated sequence consisting of mesh files displayed at a pre-specified rate, e.g., 24 mesh files per second. Motill is resource intensive and time consuming. For an animated sequence that is displayed repeatedly, it is more efficient to store Motill’s computed flight paths in a file once and read the file for each repeated display. A file format to store these flight paths is described in [3]. This study presents a conceptual data model for its file format. The conceptual model presented here is different because it captures information and knowledge. It strives to maintain semantics and facilitate content-based queries.

Conceptual models are recognized as an important tool in several computing disciplines including databases, multimedia, and software engineering. The Entity-Relationship (ER) data model used in our study dates back to the seminal work of Chen [5] and instructed by numerous computer science textbooks, e.g., [19]. Conceptual models and the idea of data transformed to information and knowledge is present in diverse scientific applications including Earth Sciences [17, 18].

3 CORE CONCEPTUAL DATA MODEL

The core data model strives to provide a two-way interaction between the user and the display, enriching the user’s overall experience. Each FLS is a computer with the capability to execute algorithms that understand and interpret the data illuminated by its swarm. One may conceptualize a swarm as one or more server racks of today’s data centers that facilitates rich interactions with the user. Backed by a conceptual data model, the FLS display becomes more than a one-way channel for visual and haptic rendering. It enables its users to inquire about the materialized objects, annotate

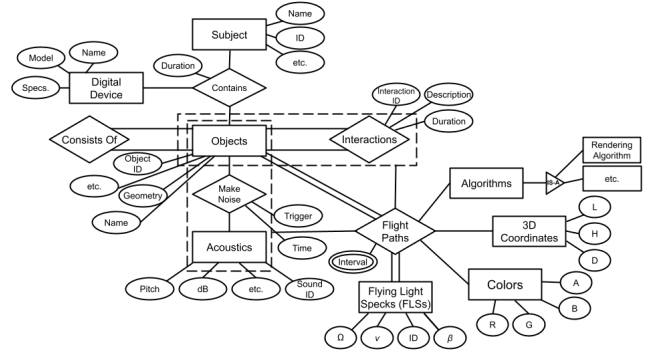


Figure 2: An extensible core conceptual data model.

and manipulate them, and provide summaries. Through these interactions, users may contribute additional data, information, and knowledge to the multimedia repository. See Figure 1.

The resulting conceptual data model consists of entity sets (rectangles) and relationship sets (diamonds). Both may have attributes, represented as ovals. Multiple entity sets may be involved in a relationship set. The relationship set is represented as a diamond. A single line from an entity set to a relationship set suggests an entity may exist without participating in a relationship. For instance, the connection between the Object and Make Noise indicates that an object may exist without acoustics. Conversely, a double line signifies mandatory participation, meaning an entity can only exist if it participates in that relationship. For example, the linkage between Flight Paths and Objects dictates that an entity within the Objects entity set must establish a relationship with Flight Paths¹. One may use an aggregated entity set (dotted rectangle) to define a relationship as an entity that may participate- in another relationship. An entity set may also participate recursively with itself through a relationship set as shown between the Objects and Consists-Of relationship sets.

Figure 2 shows a Subject entity set that contains Objects. This relationship has a Duration attribute that specifies when an object enters and exits the subject’s realm relative to the start of the subject. The Objects entity set contains the attributes Object ID, Name, Geometry, etc. The attributes of the Objects entity set are also connected to the Acoustics entity set by the Make Noise relationship set. The Make Noise relationship set includes two key attributes: Time and Trigger. These serve as indicators for the timing and initiation of sound emissions from an object. The Time attribute specifies the exact moment in a visual sequence when an object’s acoustic element should be activated. The Trigger attribute identifies the specific type of user-object interaction that should initiate the sound. For instance, if a user pets a virtual cat or pats a virtual dog, the Trigger would initiate the respective sounds of a purr or a bark. The Acoustics Entity set stores the attributes of each noise an object makes with the attributes: Sound ID, pitch, dB, Frequency, and any other attributes needed to achieve the full depth of the audio. This relationship between the Objects entity set and the Acoustics entity set is aggregated as one entity set denoted

¹This is justified because we are modelling data for an FLS display and the flight path of FLSs to illuminate the objects.

with a dashed rectangle that participates in the relationship set Flight Paths denoted with a single line.

The Objects entity set also has a recursive relationship through the Consists-Of and the Interactions relationship sets. The Consists-Of relationship illustrates how each object in the Objects entity set can be composed of smaller objects, which in turn may be made up of even smaller objects, and so forth. The Interactions set illustrates a recursive relationship in which an object may interact with another object, which in turn may interact with yet another, and so on. The Interactions relationship set has the attributes: Interaction ID, Duration, Description, etc. This relationship between the Objects entity set and the Interactions is aggregated as its own entity set denoted with a dashed rectangle that participates in Flight Paths. This many-to-many relationship set requires the total participation of the Objects and the FLSs entity sets. The FLSs entity set outlines the velocity model of an FLS (v), the duration of its flight time on a fully charged battery (β), the amount of force needed by the FLS to simulate a certain level of resistance (N), and the battery's charging time (Ω) [10]. ID is the unique identifier of an FLS. The relationship between Objects, Interactions, and Flight Paths is a fundamental aspect of this model. Each object generates its own Flight Paths, which are then adjusted based on the interactions among the objects. Similarly, when users modify these Flight Paths, it directly influences the Objects and Interactions Entity Sets. This process ensures that the model adapts to user modifications, creating a more tailored and effective system.

The Flight Paths relationship set engages either the 3D Coordinates and Colors entity sets, the Algorithms entity set, or both. This flexibility accommodates various scenarios: the digital device may supply data that directly populates the 3D Coordinates and Colors sets; alternatively, the Algorithms entity set alone may suffice to calculate the coordinates and colors of the FLSs. In some scenarios, both may work in conjunction to generate comprehensive flight paths.

The 3D Coordinates entity set specifies the Length L , Height H , and Depth D positioning for an FLS. The Colors entity set determines the brightness levels of various lights that the FLS is tasked to display. Figure 2 employs the Red, Green, Blue, and Alpha (RGBA) color scheme. The Algorithms entity set stores the algorithms necessary for computing the flight paths of each FLS. As each FLS has its own computational capabilities, this entity set utilizes the Objects, Interactions, FLSs, and other potential entity sets as inputs to generate each individual flight path on-board. The Algorithms entity set also branches, through the Is-A relationship, into a subclass, specifically for Rendering algorithms, which handles the dynamic color changes for each FLS. This Algorithms set is designed to be extendable, capable of incorporating a wide range of algorithms used for FLS manipulation. These algorithms are responsible for the calculations necessary for the bidirectional interactions between users and the display. By incorporating algorithms for lighting properties, stiffness, or the general behavior of different states of matter, the FLS flight paths can adapt to the inputs received by users when they feel, bend, or poke displayed objects for example. Depending on the specific application of our ER model, the Algorithms entity set must be specialized to include functions tailored to the task at hand. These specialized functions could continue to use the Objects,

Interactions, FLSs, and other entity sets as inputs, adapting to any changes in their attributes.

Flight Paths also has an Interval attribute which describes how long an FLS renders a color at a 3D coordinate. The Interval attribute is a multi-valued (double ellipsoid) because an FLS may be required to render the same color and 3D coordinate at different times. An example is a hummingbird beating its wings where the same positions and colors may be required for multiple flaps in place.

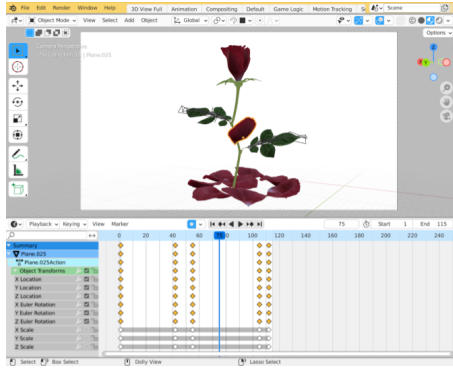
The ER diagram of Figure 2 consists of two parts, one that describes the content of an application and a second that describes how to display the content using an FLS display. The next two sections adapt this foundational abstract ER diagram to suit our two diverse applications: entertainment and healthcare.

4 ENTERTAINMENT: 3D AUTHORING TOOLS

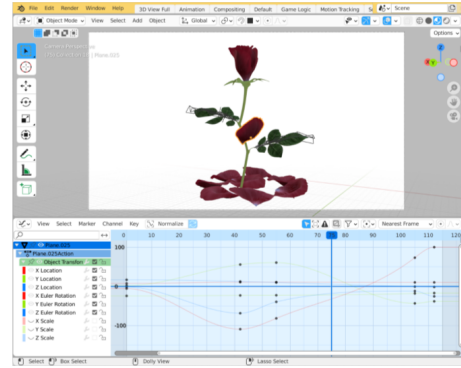
To integrate this ER model with existing animation authoring tools like Maya [6], Blender [20], ZBrush [15], and others, we observe that these tools follow a similar progression from data to knowledge. Our ER model facilitates this progression, culminating in an FLS display. In traditional animation workflows, graphic artists rely on a keyboard, a mouse, and one or more 2D monitors to create 3D animations. These authoring tools export files that capture a wealth of metadata about the animated scene. Our Entity-Relationship (ER) model can take this rich data to generate flight paths for the FLS display, elevating the viewer's experience to an immersive 3D realm. But the innovation doesn't stop at viewing; the FLS display, backed by the ER model, transforms into an interactive animation studio. Graphic artists may manipulate the 3D objects in the scene directly, effectively turning the FLS display into both a viewing and an authoring tool. This opens up new dimensions of creativity, for example, allowing graphic artists to sculpt FLS-animated "clay" into complex figures, while also labeling objects, interactions, and extending them with acoustics. The result is a more dynamic, interactive, and enriched animation process.

Today's practices: Keyframing is a technique to create life-like animations expeditiously using interpolation algorithms. The idea is for a graphic artist to specify a sequence of key frames while the algorithms compute the transitions that change one frame to the next. This process starts with a graphic artist defining keyframes at critical points along a timeline, which dictate the position and orientation of the object being animated. For example, with the rose petal animation, depicted in Figure 3a, the graphic artist defines the first keyframe that set the petal's starting position and the last keyframe that places it on the ground. Additional keyframes can be inserted in-between to introduce nuances like twists and curves in the petal's falling path. In the bottom pane of Figure 3b, these 5 keyframes are represented by the 5 vertical lines of diamonds.

An authoring software such as Blender may offer a graphic artist a choice of interpolation algorithms such as e.g., linear or Bezier, to govern how animated properties change over time. Once the keyframes and interpolation types are set, the software calculates the values of animated properties, like position and rotation, over the specified time.



(a) Keyframing.



(b) Bezier Interpolations.

Figure 3: Blender.

4.1 Advantages of a Conceptual Data Model for Authoring Tools

The Algorithms entity set of Figure 2 describes the keyframe-based approach. A swarm of FLSs corresponding to the changing portion of a keyframe may execute the algorithm to compute their flight paths. Each object in the FLS display retains its full set of individualized attributes, interactions, and compositional elements, allowing users to query these objects. For example, a user may issue the following query, “Who is hiding in the closet?” and the system may accurately respond and even use the FLS display to illuminate the hidden entity.

The proposed conceptual model aligns more naturally with human cognitive processes, particularly our innate spatial awareness. Humans naturally perceive the world in three dimensions, and our ability to understand spatial relationships is finely tuned. Animating 3D objects in a 3D space simulated with FLSs taps into this natural aptitude, making it easier and more intuitive to understand complex scenes. This also allows multiple graphic artists to visualize the same display from different angles and simultaneously edit and mold the objects within it. Any changes made are captured and updated in the conceptual model, offering a dynamic, interactive, and adaptable animation environment. In this collaborative setting, character designers, graphic artists, sound designers, and other creators may collaborate in real-time within the same 3D space, enhancing the creative process and allowing for immediate feedback and iterative design.

4.2 Extending the Core Model for Animation

We use an ontology to tailor the core model of Figure 2 to generate the extended ER diagram of Figure 4. The Subject entity set is renamed to the Scene entity set, maintaining attributes Name and ID, and can be expanded to include any other key identifiers deemed essential for both viewing and editing the animation.

Similarly, the Digital Device entity set is named the Authoring Tool entity set, representing software platforms like Autodesk Maya, Blender, Zbrush among others. This updated entity set provides an understanding of the tools used to create animations and includes attributes like Name, Model, and software specifications.

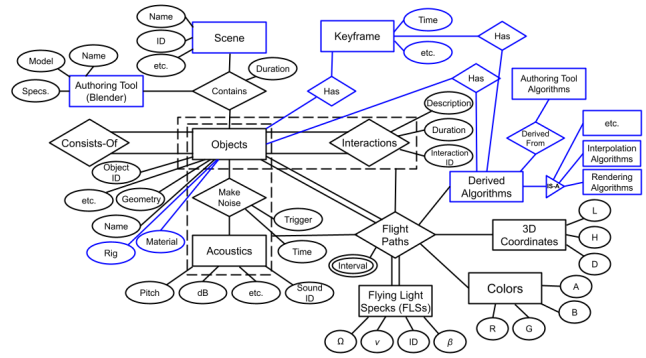


Figure 4: ER Model for Animation Tools

The Objects entity set is enhanced, now with attributes: Object ID, Name, Material, Rig (the “skeletal” structure of an animated object), and Geometry.

The entity set originally named Algorithms has been renamed to Derived Algorithms. This set is derived via a new relationship set called Derived From, which links it to a new Authoring Tool Algorithms entity set. The latter contains algorithms employed by the 3D animation authoring software for its own simulations, which serve as the foundation for the Derived Algorithms now used to calculate the flight paths.

The Derived Algorithms entity set also branches, through the Is-A relationship, into subclasses, specifically for Rendering and Interpolation algorithms. The Rendering Functions handle the dynamic color changes for each FLS, while the Interpolation Functions facilitate the seamless transition between keyframes whenever a property of an object is changing over time. The Derived Algorithms entity set is designed for flexibility and scalability, enabling the integration of a diverse array of animation algorithms. An illustrative example is Blender’s sophisticated ray tracing algorithms, which handle optical phenomena like refraction, reflection, transparency, and absorption. This entity set can incorporate similar ray tracing algorithms, reimaged for 3D displays as opposed to Blender’s 2D framework. Additionally, it may encompass a range of general

functions that model various physical states of matter, providing a comprehensive toolkit for accurately simulating real-world physics in animation.

The Objects entity set now branches through a new Has relationship set to a newly introduced Keyframe entity set. The Keyframe entity set includes a Time attribute and is designed to be extendable, for example, to allow for the addition of further attributes to specify which properties of the object are undergoing changes in a particular keyframe. The Time attribute marks the specific moment the keyframe occurs in the animation. Furthermore, the Keyframe entity set is linked to the Derived Algorithms set through another Has relationship set, specifying the interpolation algorithm used for that particular keyframe. Similarly, the Objects entity set maintains a Has relationship with the Derived Algorithms set to access the required Rendering Algorithms for dynamically coloring the object.

The remaining foundational elements of the original ER model remain intact, ensuring that the adapted model is both robust and tailored to the specific queries needed for animation. In this model, each object in a scene is an entity in the Objects entity set, complete with its geometry, material, geometry, rig, and keyframes for any property changes. For instance, to depict a position change in a rose petal, multiple keyframes would be required and stored in the Keyframes Entity set and related through the Has relationship set to the rose petal's entry in the Objects entity set. The Interactions and Consists Of entities can be populated using an interpretation of the 3D rendering with an algorithm from the Derived Algorithms set, or can be manually annotated by the viewer or graphic artist. The Algorithms entity set would be populated with all the functions needed to take the geometry, material, rig, and keyframes of an object, combined with the objects it consists of, its interactions, and the sounds it makes, to derive a flight path for every FLS emulating that object.

5 HEALTHCARE: MRI SCANS

Magnetic Resonance Imaging, MRI, scanners create 3D images of the anatomical structure of the human body. These images are used for both research and diagnosis of an illness or disorder. There are two types of scans, structural and functional MRI (fMRI). A structural scan is a single 3D image represented as a matrix of voxels. A voxel corresponds to measurements from a small volume of body, quantifying a property such as the volume's (tissue) density or water content. fMRI is a sequence of structural images as a function of time, describing the change in the tissue. A popular use of fMRI is to analyze brain activity overtime and in response to a stimulant.

An immersive FLS display that illuminates either a structural or an fMRI scan may be the size of a room. A physician may enter this room with an organ illuminated in front of them. They may step (zoom) into the organ by having it expand around them to analyze individual voxels visually. The physician may touch the voxels to experience their stiffness (this stiffness would be derived from the properties observed in the MRI cross referencing an external database that stores the necessary data to simulate an accurate relative stiffness and density). The physician may shrink the rendering as appropriate to orient themselves relative to an organ before zooming (stepping) into it to examine its voxels. The physician may annotate an organ, a voxel, or groups of voxels. The physician may

color code voxels to highlight a disease, and associate knowledge (e.g., identify a spreading cancer) with different organs, tissue and one or more voxels.

A conceptual model for the structural and fMRI scans enables the aforementioned interactions with an FLS display. Acting as the 'brain' of the system, the model enables a more sophisticated interaction with the medical images. Physicians can go beyond basic observational interactions and engage with a true-to-life 3D representation of various anatomical structures. This enhanced interaction allows for a more comprehensive and nuanced analysis, facilitating the generation of in-depth diagnostic reports. Moreover, the FLS display, informed by the ER model, becomes an intelligent visualization tool that 'understands' what it is displaying. This not only enriches the experience for the physician but also provides an interface where health care providers can inquire about interactions shown and object's properties.

Today's practices: Current MRI practices predominantly generate 2D images in single planes, often referred to as "slices." This 2D display format inherently limits the level of spatial comprehension, particularly for complex medical procedures, for example the removal of a Myxoma [1], an inner heart tumor. Specifically the 2D framework causes a diminished depth perception in traditional MRIs. This limitation significantly hampers the ability to visualize complex structures such as white matter tracts within the nervous system, which are typically assessed using Diffusion Tensor Imaging (DTI), a specialized form of MRI. Efforts to overcome the spatial awareness limitations inherent in 2D medical imaging have led to the exploration of augmented and virtual reality technologies. For example, in one study [8] researchers employed Virtual Reality (VR) head-mounted displays as a solution to enhance the visualization and labeling of lesions in brain scans. However, a primary drawback of this technology is VR-induced motion sickness, which can cause nausea in users after just 5 to 10 minutes of usage.

Another limitation of today's MRI visualization tools is the absence of tactile engagement. Currently, physicians rely on T1-weighted and T2-weighted MRI images to discern various tissue types and conditions. T1-weighted images visualize anatomical details and fat. T2-weighted images are more sensitive to fluid and useful for detecting inflammation or edema. Although physicians can adjust monitor settings to better interpret these grayscale images, they are limited to visual observation. An FLS display will enable physicians to touch and poke the illuminated tissue to experience its density and stiffness directly.

At the time of this writing, the field of MRI imaging also faces inconsistencies in data presentation conventions. For instance, radiologists often view images with the brain's right side on the image's left, known as the "radiological convention," aligning with the view of the body from the foot of the bed. In contrast, neurologists prefer the "neurological convention," where the brain's left side matches the image's left. This inconsistency in brain imaging means that the interpretation of the X dimension is always a concern. Given the brain's left-right symmetry, discerning an image's orientation becomes challenging without context.

5.1 Advantages of a Conceptual Model for MRI Data

The current approach to displaying and interacting with medical images can be significantly enhanced by integrating a conceptual model with an FLS display. Such a display, enriched with detailed data, allows physicians to make more informed queries about the images. For instance, both radiologists and individuals with limited medical expertise could ask questions about constricted blood flow in the brain or the degree of inflammation in the heart since a patient’s last MRI scan.

Moreover, this system enables tactile interaction with the 3D images. By incorporating data from CT scans—which also use voxels and are thus compatible with our conceptual model—we can accurately simulate the geometry, relative density, and other properties of each organ. This information is stored in the ER model and informs the flight paths and haptic capabilities of the FLSs, making kinesthetic interaction possible. For example, a doctor can brush his hand over a simulated brain and feel the changes in tissue to identify if a tumor is forming, and how the structural integrity of the organ will change if it is removed.

While AR and VR technologies offer a 3D experience, they come with limitations such as user nausea and motion sickness during extended use [8]. On the other hand, the FLS system materializes objects in a true 3D space, eliminating these issues. It undoubtedly provides a more comprehensive and easier-to-understand view, enabling physicians of various backgrounds and specialties to collaborate on a single patient’s scans.

The field of MRI imaging is fraught with inconsistencies in data presentation, varying between radiologists and neurologists, among others. These inconsistencies can make it challenging to interpret MRI scans accurately and efficiently. Another advantage of our Entity-Relationship model is that it proposes a standardized approach to representing MRI data, aiming to unify these divergent practices and offer a consistent, 3D, easy-to-analyze format that can be universally adopted for MRI scans.

5.2 Extending the Core Model for MRI Scans

Figure 5 shows extensions of the core model of Figure 2 for MRI scans. The Subject entity set is now renamed to the Patient entity set with the attributes Name, ID, and any other individual identifying information that is critical to a diagnosis. The Digital Device entity set will now be a Medical Imaging Equipment (MRI) entity set, in this case an MRI entity set, with its attributes being Name, Model, and Specifications of the machine. The Objects Entity set is now adjusted to be the Organs Entity set with attributes: Organ ID, Name, Geometry, and Disease (the disease that each organ may be affected by). Disease is a double oval attribute denoting that it is a multidimensional attribute as one organ may be experiencing the symptoms of multiple diseases at once.

Our Algorithms Entity set has been extended to include Rendering and Organ Annotation Algorithms. The Rendering Algorithm is responsible for managing dynamic color changes, determining the colors of the FLSs in the Flight Paths set. Meanwhile, the Organ Annotation Algorithm processes data from MRI scans to analyze shape, size, location, signal intensity, texture, and more, thereby identifying and automatically labeling, areas of concern, tissues,

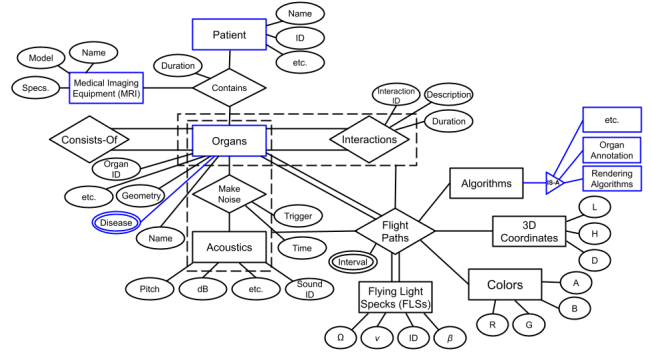


Figure 5: ER Model for MRI Data

muscles, and other organs within the MRI for targeted physician review. To accurately simulate tissue properties depicted in the MRI, the Algorithms Entity set also incorporates functions that model how organs interact with each other, including changes in interactions and alterations to their properties. For instance, if a surgeon wants to simulate the extraction of a brain tumor, the FLS must be capable of demonstrating the structural integrity of the surrounding tissue to indicate whether it could collapse. This is just one example; the algorithms required to simulate the properties and interactions of all bodily tissues are likely to be extensive.

The population of this Entity-Relationship diagram diverges from that of the authoring tool diagram. Unlike authoring tools, the MRI data primarily populates the 3D Coordinates and Colors Entity sets. These sets serve as inputs for the Flight Paths Relationship set and, crucially, the Algorithms Entity set. The Algorithms set, in turn, enriches the Organs Entity set along with its associated relationships and entities. Manual annotations by physicians can further augment the Objects, Interactions, and Consists-Of Entity sets, enhancing their accuracy and comprehensiveness. It’s worth noting that the ER model can also be populated using data from other medical imaging techniques, such as CT scans, PET scans, and echocardiography for acoustic information. This allows for a more comprehensive and multi-modal approach to medical imaging.

6 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

This paper described a conceptual model for intelligent 3D multimedia data rendered using FLS displays. The data is intelligent because it includes summaries in the form of information and knowledge, facilitating content-based retrieval. A user may issue a query, illuminate its results using the FLS display, and annotate it with additional information and knowledge as necessary.

An immediate research direction is to develop a logical and physical counter-part of the conceptual model for Blender, a 3D authoring tool. At its simplest form, the physical implementation enables Blender to output flight path of FLSs by computing coordinates and colors. Its algorithms may use Motill [10] as their starting point and reduce its complexity by using the metadata (information and knowledge) provided by a graphic artist to Blender. We will investigate the scalability and efficiency of this data model. We

will also investigate content-based queries described in Section 4 and use of Blender’s keyframing algorithms to compute FLS flight paths.

7 ACKNOWLEDGMENTS

This research was supported in part by the NSF grant IIS-2232382.

REFERENCES

- [1] Khamidulla Abdumajidov and Khairulla J Buranov. 2023. Giant Myxoma of the Right Sections of the Heart Obstructing the Inferior Vena Cava: A Case Report. *Heart, Vessels and Transplantation* Articles Published Ahead of Print (2023).
- [2] Hamed Alimohammadzadeh, Rohit Bernard, Yang Chen, Trung Phan, Prashant Singh, Shuqin Zhu, Heather Culbertson, and Shahram Ghandeharizadeh. 2023. Dronevision: An Experimental 3D Testbed for Flying Light Specks. In *The First International Conference on Holodecks* (Los Angeles, California) (*Holodecks '23*). Mitra LLC, Los Angeles, CA, USA, 1–9. <https://doi.org/10.61981/ZFSH2301>
- [3] Hamed Alimohammadzadeh, Daryon Mehraban, and Shahram Ghandeharizadeh. 2023. Modeling Illumination Data with Flying Light Specks. In *ACM Multimedia Systems* (Vancouver, Canada) (*MMSys '23*). Association for Computing Machinery, New York, NY, USA, 363–368. <https://doi.org/10.1145/3587819.3592544>
- [4] Sean Braley, Calvin Rubens, Timothy Merritt, and Roel Vertegaal. 2018. Grid-Drones: A Self-Levitating Physical Voxel Lattice for Interactive 3D Surface Deformations. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) (*UIST '18*). Association for Computing Machinery, New York, NY, USA, 87–98. <https://doi.org/10.1145/3242587.3242658>
- [5] Peter P. Chen. 1975. The Entity-Relationship Model: Toward a Unified View of Data. In *Proceedings of the International Conference on Very Large Data Bases, September 22-24, 1975, Framingham, Massachusetts, USA*, Douglas S. Kerr (Ed.). ACM, 173. <https://doi.org/10.1145/1282480.1282492>
- [6] Dariush Derakhshani. 2012. *Introducing Autodesk Maya 2013*. John Wiley & Sons.
- [7] Joseph DeSimone. Onstage at TED2015. See https://www.ted.com/talks/joseph_desimone_what_3d_printing_was_100x_faster?language=ent-20146. What if 3D Printing was 100x Faster?
- [8] Dominique Duncan. 2023. Large-scale Data Archives and AI methods to Study Neurological Disorders, Ming Hsieh Department of Electrical and Computer Engineering Conferences, Lectures, & Seminars, September 12, 2023.
- [9] Shahram Ghandeharizadeh. 2021. Holodeck: Immersive 3D Displays Using Swarms of Flying Light Specks. In *ACM Multimedia Asia* (Gold Coast, Australia). ACM, New York, NY, USA, 1–7. <https://doi.org/10.1145/3469877.3493698>
- [10] Shahram Ghandeharizadeh. 2022. Display of 3D Illuminations using Flying Light Specks. In *ACM Multimedia* (Lisboa, Portugal). ACM, New York, NY, USA. <https://doi.org/10.1145/3503161.3548250>
- [11] Shahram Ghandeharizadeh and Luis Garcia. 2022. Safety in the Emerging Holodeck Applications. In *CHI 2022 Workshop on Novel Challenges of Safety, Security and Privacy in Extended Reality*.
- [12] Seth C. Goldstein, Jason D. Campbell, and Todd C. Mowry. 2005. Programmable Matter. *Computer* 38, 6 (2005), 99–101. <https://doi.org/10.1109/MC.2005.198>
- [13] Antonio Gomes, Calvin Rubens, Sean Braley, and Roel Vertegaal. 2016. Bit-drones: Towards using 3d nanocopter displays as interactive self-levitating programmable matter. In *Proc. ACM CHI Conference on Human Factors in Computing Systems*. 770–780.
- [14] Adrian Haarbach, Tolga Birdal, and Slobodan Ilic. 2018. Survey of Higher Order Rigid Body Motion Interpolation Methods for Keyframe Animation and Continuous-Time Trajectory Estimation. In *2018 International Conference on 3D Vision (3DV)*. 381–389. <https://doi.org/10.1109/3DV.2018.00051>
- [15] Eric Keller. 2011. *Introducing ZBrush*. John Wiley & Sons.
- [16] William A. McNeely. 1993. Robotic Graphics: A New Approach to Force Feedback for Virtual Reality. In *Proceedings of IEEE Virtual Reality Annual International Symposium*. 336–341. <https://doi.org/10.1109/VRAIS.1993.380761>
- [17] Hampapuram K. Ramapriyan, David Isaac, Wenli Yang, and Steve Morse. 2005. Large Scale Data Mining to Improve Usability of Data - An Intelligent Archive Testbed. In *IEEE International Geoscience & Remote Sensing Symposium, IGARSS 2005, July 25-29, 2005, Seoul, Korea, Proceedings*. IEEE, 5618–5621. <https://doi.org/10.1109/IGARSS.2005.1526050>
- [18] Hampapuram K. Ramapriyan, Gail McConaughy, Christopher Lynnes, R. Harberts, L. Roelofs, Steven Kempler, and Kenneth McDonald. 2002. Conceptual Study of Intelligent Data Archives of the Future. In *Tenth Goddard Conference on Mass Storage Systems and Technologies in cooperation with the Nineteenth IEEE Symposium on Mass Storage Systems*. 67–74. <https://doi.org/10.1109/VR.2007.352465>
- [19] Abraham Silberschatz, Henry Korth, and S. Sudarshan. 2020. *Database System Concepts. Chapter 6*. (7 ed.). McGraw Hill, USA.
- [20] Oliver Villar. 2021. *Learning Blender*. Addison-Wesley Professional.