# ViRMA: Virtual Reality Multimedia Analytics at LSC 2021

Aaron Duane
IT University of Copenhagen
Copenhagen, Denmark
aadu@itu.dk

Björn Þór Jónsson
IT University of Copenhagen
Copenhagen, Denmark
bjth@itu.dk

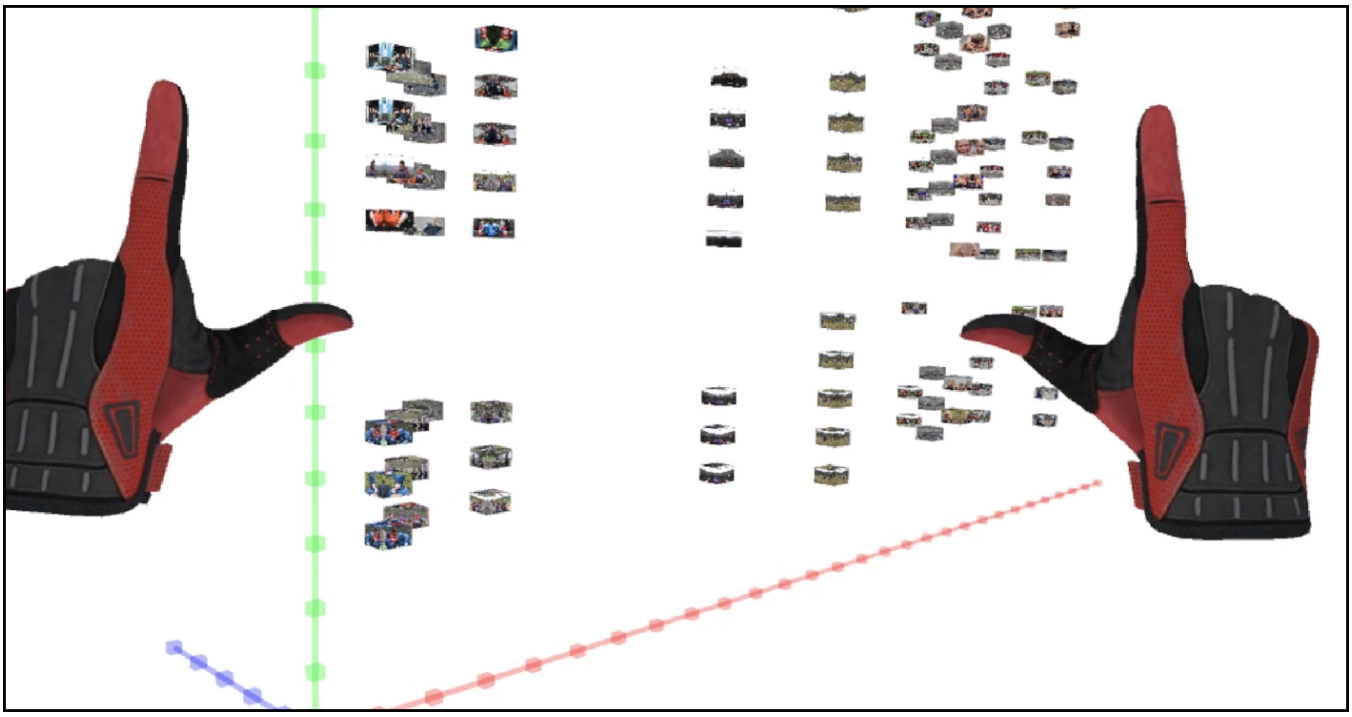


**Figure 1: A snapshot of a user's browsing state in ViRMA's projection space**

## ABSTRACT

In this paper we describe the first iteration of the ViRMA prototype system, a novel approach to multimedia analysis in virtual reality and inspired by the $M^3$ data model. We intend to evaluate our approach via the LSC to serve as a benchmark against other multimedia analytics systems.

## CCS CONCEPTS

• **Information systems** → **Users and interactive retrieval**; • **Human-centered computing** → **Virtual reality**.

## KEYWORDS

lifelogging, virtual reality, human-computer interaction, multimedia analytics

**ACM Reference Format:**
Aaron Duane and Björn Þór Jónsson. 2021. ViRMA: Virtual Reality Multimedia Analytics at LSC 2021. In *Proceedings of the 4th Annual Lifelog Search Challenge (LSC '21), August 21, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3463948.3469067

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*LSC '21, August 21, 2021, Taipei, Taiwan*
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8533-6/21/08…$15.00
https://doi.org/10.1145/3463948.3469067

## 1 INTRODUCTION

In this paper we present the first iteration of the ViRMA (**Vi**rtual **R**eality **M**ultimedia **A**nalysis) prototype which we intend to evaluate via the LSC competition at ACM ICMR 2021. The motivation for the ViRMA system arose from the confluence of three notable trends in computing: emphasis on interactive analysis; availability of novel access mechanisms; and increasing collection scale.

In recent years, increasing emphasis has been placed on interactive analysis by users in various multimedia application domains, as it has become clear that to satisfy diverse and dynamic information needs, effective collaboration between human and machine is

necessary [5]. This calls for combining sophisticated multimedia analysis, scalable data management, and interactive visualisation into a single system that supports the user's interactive analysis of a media collection [8].

At the same time, hardware for interacting with data in virtual reality has been improving rapidly, and has reached the point where quality interactions are possible with affordable hardware. Past research has suggested that VR is highly valuable due to its immersive quality, the degree to which it projects stimuli onto the sensory receptors of users, and that it will lead to more natural and effective human-computer interfaces [6].

With regard to increasing collection scale, it is clear that much of the phenomenal growth in data production and storage in recent years has been in the form of media files, as evidenced by the rise of media streaming and sharing platforms such as Netflix and YouTube. Today, any person, not just a lifelogger, is capable of producing hundreds of thousands of image or video files. The need for scalable systems for analysing large media collections is therefore clear.

The foundation for the ViRMA prototype is the $M^3$ model [2] which is in turn based on the merging of concepts from business intelligence, such as analytical processing (OLAP), multidimensional analysis (MDA), and faceted browsing. Though ViRMA is not the first VR lifelog application to compete in the LSC [1], it is the first VR lifelog application to utilise a visualisation paradigm which relies explicitly on the effective representation of multimedia data in 3D virtual space. With the ViRMA prototype, we aim to consider the impact of collection scale on the VR interfaces to multimedia analytics, and use interactive competitions such as VBS and LSC [3] as a platform to evaluate our approach.

## 2 THE M$^3$ MODEL

The M$^3$ data model (pronouced "emm-cube") considers multimedia objects to reside in a multi-dimensional metadata space, and then provides support for exploring that space via operations to project it down to 3D space. In this section, we present the multi-dimensional concepts and exploration operations, using examples from the LSC collection.

### 2.1 Multi-Dimensional Metadata Space

In the following, we first describe the key concepts of the M$^3$ model, and then briefly outline how the LSC collection is translated into the M$^3$ model.

**Media Objects** Media *objects* are the media files at the center of the data model.

**Metadata Tags** Each object is described by a set of metadata *tags*. These metadata can be (i) created with the media object, (ii) extracted from the media content (e.g., using semantic classifiers), or (iii) provided by users. Each association between a media object and a tag is called a *tagging*.

**Metadata Dimensions** Each tag resides in a particular *tagset*, which groups together semantically related tags. A tagset can then further be organised using one or more *hierarchies*, which provide a semantically meaningful structure for the tagset as a basis for exploration. A media object is considered to be tagged with a hierarchy node, if it is tagged with a tag in one of its sub-trees. Both tagsets and hierarchies are considered *dimensions*.

**Metadata Space** The cross-product of all the dimensions yields the multi-dimensional *metadata space*, which has the form of a hyper-cube in the multi-dimensional space. Each *cell* of the cross-product contains all media objects tagged by the tags that define the cell in each dimension.

Note that the metadata space is naturally sparse, but when projected down to 3D it becomes much more dense. How to achieve this projection is described next.

### 2.2 Multi-Dimensional Exploration

While media objects reside in a multi-dimensional metadata space, human navigation of this space can occur in (at most) three dimensions, referred to as the visible axes of the 3D exploration cube. The data model offers multiple operations for the purposes of such navigation.

**Projection** By associating a metadata dimension with one of the three visible axes, the collection is projected onto this axis. Each cell on this axis will correspond to one tag, and be represented by (some) media objects containing that tag. Only images containing a tag in this dimension will be represented, while images tagged with multiple tags will be represented in each corresponding cell. When more than one dimension is assigned to visible axes, only images containing tags in all those dimensions are represented.

**Slicing and Dicing** By defining a filter on a dimension, a user can focus on a slice of the multi-dimensional space, termed slicing the cube. Defining filters on multiple dimensions is, in turn, termed dicing the cube. The filters can be: (i) tag filters, which apply to a tag-set and retain only the media items tagged with that tag; (ii) range filters, which apply to tagsets with a natural order, and retain all media items tagged with any tag within the range; and (iii) hierarchy filters, which apply to a node in a hierarchy, and retain all media items tagged with any tag in any sub-tree of the node. A slicing operation can be applied both to dimensions that are associated with a visible axis and dimensions that are not currently visualised.

**Drilling Down and Rolling Up** When a hierarchy is associated with a visible axis, it is natural to display only the children of the root, where each child represents all the media objects in its sub-tree. Users will then often wish to explore one particular subtree, which is achieved by (a) assigning a new hierarchy filter to that child, and displaying its children on the visible axis. This is called *drilling down*. Repeating this action results in a trail of increasingly narrow hierarchy filters, along with visualising lower levels of the hierarchy. The opposite action, going from a child to its parents, is called *rolling up*, and is achieved by removing the last hierarchy filter and displaying the children of the parent.

**Pivoting** Once a user has explored a visible dimension and perhaps set a suitable filter, the user may desire to replace the dimension on the visible axis with some other dimension for further analysis. This is called *pivoting*, referring to changing directions in the analysis of the multi-dimensional space.

Importantly, any filters that have been set on the dimension that is replaced, will be retained until the user explicitly removes them.

The state of exploration at any time is defined by the dimensions that are currently visible, which implicitly define associated filters, as well as any other filters that the user has applied. These dimension thus define the skeleton of the current the 3D exploration cube, while the media items that pass through all these filters define the contents of the 3D exploration cube. We refer to the definition and content of the current 3D cube as the *browsing state*.

## 2.3 LSC and M$^3$

In the LSC collection, the media objects are the lifelog images, taken every 20-30 seconds, which together aim to capture the experiences of the lifelogger. The LSC metadata provides structural tags (e.g., time, date, and GPS coordinates), extracted tags (e.g., location derived from GPS coordinates), and user-provided tags (e.g., detailed location labels). In addition to this, we have extracted semantic features using the ImageNet Shuffle [4], a deep neural network using the ResNeXt-101 architecture [7]. For each of the 191K images, the 5 highest-scoring concepts are retained as tags, resulting in 6,399 different tags. Since the tags extracted by ImageNet Shuffle directly correspond to a subset of the WordNet database, we created a large hierarchy containing every distinct tag using the WordNet Python API.

In total, the hierarchy has 8,803 nodes, which means that some tags are replicated in the hierarchy. The root of the hierarchy is an *Entity* tag, which has two children: *Abstract entity* and *Physical entity*. The former covers 409 nodes, while the latter covers 8,393 nodes. Overall, the hierarchy is quite unbalanced, with occasional chains of abstract nodes with very few children each (in total, 345 nodes have a single child, and the longest path from root to leaf is 19 levels) and occasional nodes with many children (6 nodes have more than 50 children, and 2 nodes have 89 children). In its current form, the *Entity* hierarchy is unwieldy even for experienced users. The ViRMA prototype therefore implements a search feature (explained in further detail in the next section) to identify useful entry points to it. Improving the hierarchy itself, however, is an important future task.

# 3 SYSTEM DESCRIPTION

The ViRMA system is built in the Unity game engine and uses the Valve Index as its virtual reality platform. The Valve Index is part of the most recent generation of VR hardware which is available at the time of development and continues to rely on two wireless controllers like its predecessor, the HTC Vive. The controllers offer numerous interactive components such as a touchpad, haptic feedback, and pressure sensitive grips, however, for the sake of brevity, we will not describe the specific way the controllers are used for every interaction within the ViRMA system, unless such an interaction is sufficiently unique or noteworthy.

The core components of ViRMA can be described as falling under two main categories; components which support the generation of user queries, and components which support the visualisation and exploration of the data produced by these queries. We refer to these categories respectively as *query generation* and *data projection*

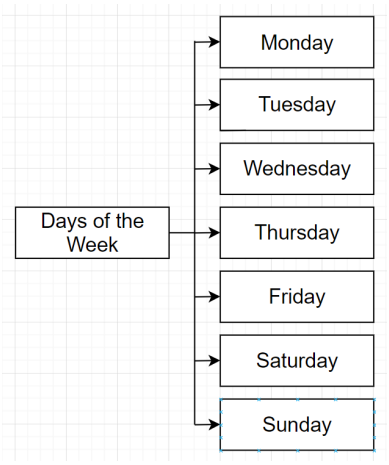



**Figure 2: Example of a tagset**

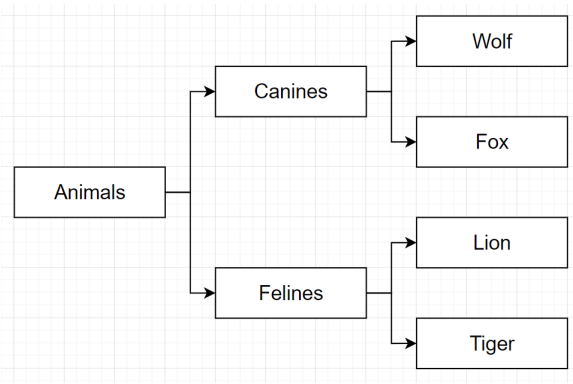



**Figure 3: Example of a hierarchy**

within the context of the ViRMA system. In this section we will describe each of the components in these categories and how the user might interact with them in order to complete a typical LSC task.

## 3.1 Query Generation

As was described earlier, the metadata associated with the LSC dataset is organised semantically into tags, tagsets, and hierarchies (see Figures 2 and 3). These serve the underlying M$^3$ model but, from the perspective of the user, these serve as potential filters which can be applied to their current query or browsing state.

It is important to note that within ViRMA, these filters can be applied to the data in two fundamental ways. The most obvious approach is to locate a tag, tagset, or hierarchy of interest, and simply apply it as a direct filter on the dataset. This will reduce the entire set of potential results to those which are tagged by that filter. In the case of a tagset or hierarchy, this will include all of their children. The other fundamental way of applying a filter in ViRMA, is to visualise it on a spatial axis in the virtual environment. We refer to this as projecting the filter as a dimension and will explore it in more detail in the *data projection* section.

With so many potential filters available in ViRMA, including the different methods by which they can be applied, it is imperative that a user can effectively navigate and browse this metadata before deciding which are most appropriate to use and how they should be applied. Browsing or searching for a tagset is comparatively straightforward as each tagset only contains a single group of semantically associated tags. Hierarchies, however, can contain any number of tags at varying depths of association and can quickly become cumbersome to navigate when searching for a specific tag or tagset in the hierarchy. One option is to restrict the depth and complexity of hierarchies so they are easier to navigate, but this reduces the utility of the hierarchy once it is projected as a dimension in the virtual space. For example, the user will have less opportunities to *drill down* or *roll up* and, the now shallower but broader hierarchy depths, produce too much granularity when *sliced and diced*, resulting in excessive cognitive burden.

To maintain the benefits of more semantically detailed hierarchies and yet mitigate their associated detriments, we introduced the concept of a *dimension explorer* to the ViRMA system which we will now describe.

## 3.2 Dimension Explorer

The dimension explorer is a dedicated user interface element which can be opened at any time within the virtual space. Once loaded, it is populated with a list of all tagsets and all hierarchies (at their topmost level) which are currently available for the dataset. From this list, the user can drill down into any tagset or hierarchy and their children before deciding to apply them as either a direct filter or project them as a dimension in the virtual space. When viewing the contents of a hierarchy in the dimension explorer, the user's depth in the hierarchy is contextualised by also displaying it's parent and children, if any exist.

As has been noted, in situations with deep or complex hierarchies, this approach can become tedious when the user cannot locate a specific tag or tagset they are searching for. To address this, the dimension explorer contains a search input which enables the user to search for specific tags or tagsets within their respective hierarchies. Selecting this search input will open a virtual keyboard which the user can interact with inside the virtual space using their wireless controllers. By submitting whole or partial tag names, the dimension explorer is loaded with any matching or related tags and tagsets whilst continuing to visually preserve their context within the hierarchy. This empowers the user to select whatever depth of a hierarchy that is appropriate for their current query or browsing state to be applied as a direct filter or to be projected into the virtual space.

## 3.3 Data Projection

Now that we have established the primary interaction mechanism by which a user can apply filters to the target dataset, we can begin to explore how this data is visualised within ViRMA's virtual environment. As previously stated, we refer to the concept of applying a filter as a visible dimension in the virtual space as *data projection*. This is to draw a distinction between data or metadata which can be visualised as a dimension on a spatial axis in the environment and data or metadata which can be visualised in other contexts, such as within the dimension explorer. If we consider all the data and metadata organised by the $M^3$ model as existing in a multi-dimensional space, projecting dimensions in the ViRMA system is the equivalent of taking a specific slice of that multi-dimensional space and mapping it to the three spatial dimensions available in our virtual environment.

*3.3.1 Slicing and Dicing.* When the user first loads into the ViRMA system, the virtual environment is empty and the only interactive elements present are those necessary to support initial query generation, such as the dimension explorer. We refer to the virtual environment within ViRMA as the projection space.

Upon selecting a filter the user wishes to project as a dimension, and choosing which axis the user wants to map the dimension to (i.e. X: left/right, Y: up/down, or Z: in/out), the projection space is populated with the relevant axis and a representation of the data along that axis. For example, if the user were to project the tagset "days of the week" to the X axis, this would result in the axis being populated with 7 cells labelled with each day of the week. At present, each cell is represented in the projection space as a cuboid with a preview of the image data associated with it displayed on each side (see Figure 1). With no other active dimensions applied, each of these cells would represent every image in the dataset captured on each day of the week. In the context of the $M^3$ model, this is referred to as *slicing* the multi-dimensional space.

Though one projected axis can be helpful for some queries, it is more likely that the user will need to project other dimensions to the remaining axes in order to better refine their query. For example, with the "days of the week" tagset still on the X axis, the user might project the "alcohol" tagset to the Y axis. The "alcohol" tagset is located in the "beverages" hierarchy and contains tags such as "beer", "wine", and "spirits". Now the projection space will repopulate with new cells representing the days of the week along the X axis and the various alcohol tags along the Y axis. In the context of the $M^3$ model, projecting more than one dimension is referred to as *dicing* the multi-dimensional space and, from the perspective of the user, provides a data representation that conveys groups of images containing the various alcoholic drinks on each day of the week. The user can continue slicing and dicing the multi-dimensional space by projecting to the remaining third spatial dimension with any appropriate filter that suits their query.

*3.3.2 Drilling Down and Rolling Up.* When a tagset that exists as part of a hierarchy is projected to an axis, the user has the additional option to *drill down* or *roll up*. In the context of drilling down, this would involve drilling into a child of the current tagset and repopulating the projection space with that child's children on the axis instead. For example, with the "alcohol" tagset, a user might want to drill into "spirits" on the axis. The "spirits" tagset might contain children such as "whiskey", "vodka" or "rum" and upon drilling down, they would replace the parent tagset which was previously applied to the axis.

For rolling up, this naturally produces the opposing effect and involves re-populating the axis with the parent of the current tagset, and subsequently any siblings of that parent on that level. For example, if we rolled up from the children of the "alcohol" tagset, we would populate the axis with the "beverage" tagset, and will see "alcohol" and any of its siblings, such as "soda", "tea" or "coffee"

now on the axis. Drilling down and rolling up can be accomplished easily in the ViRMA system by targeting a specific axis with one of the wireless controllers and selecting the appropriate contextual action which is displayed to the user.

*3.3.3 Pivoting.* Once a dimension has been applied to an axis in the projection space, it is important to understand that this dimension can be removed from the axis in two fundamentally different ways. The first method is the user can simply clear the dimension entirely, and all filters that were associated with the dimension are removed from the browsing state, increasing the amount of potential results in the current query. The second method is to replace the dimension on an axis whilst maintaining that dimension's filters on the browsing state. This is referred to as *pivoting* in the context of the $M^3$ model and is the equivalent of using the dimension explorer to apply a number of tags or tagsets as direct filters before projecting a different tagset as a dimension to one of the axes in the projection space. It is imperative that this concept is effectively conveyed to the user in the ViRMA system as it is fundamental in the user's understanding of the current browsing state within the projection space.

*3.3.4 Cell and Timeline Exploration.* Once the user has sufficiently refined their query using the aforementioned techniques, it is likely that they will want to explore the image contents of an individual cell in the data projection space. This can be accomplished at any time by pointing one of the wireless controllers at the relevant cell and selecting the contextual option that appears. This will temporarily reload the projection space with all of the images contained in that cell. Furthermore, While browsing a cell's contents, if the user wishes to view any individual result in the context of the wider lifelog, they can select the image via a contextual interaction and it will load a timeline above the cell's contents displaying the image as it appeared in the lifelog. The user can then scroll left in this list to move backwards in time and right to move forwards in time. Finally, the user may return to the original projection space by selecting the appropriate button on either controller.

*3.3.5 Navigation.* Now that we have examined all of the data projection components which explicitly map to concepts related to the $M^3$ model, we can finish by discussing some of the interaction components which are unique to virtual reality as a platform. First and most important is how the user navigates the virtual space. This can be accomplished in two primary ways. The first, and most direct way, is for the user to physically move around the space themselves, however this is obviously restricted by the physical space available to the user where the virtual reality system is set up. The second way a user can navigate the projection space is to pan and rotate the entire space with their wireless controllers. The can be accomplished by pressing the appropriate control on one of the wireless controllers and gesturing in a specific direction.

## 3.4 LSC Task Walkthrough

Now that we have provided an overview of the ViRMA system, we will attempt to describe how the system might be used to approach solving an LSC-style task. Below we have taken each segment of a past LSC task and described what actions might be taken in the 30 seconds before the next part of the task is revealed to the user.

> *A red car beside a white house.*

The user begins by using the dimension explore to project tagsets and hierarchies such as "vehicles", "buildings" and "colours" to various axes in the projection space, drilling down and rolling up where appropriate. Since they are unlikely to be confident of a single image at this stage in the task, the user takes the opportunity to browse the results and familiarize themselves with the dataset.

> *A red car beside a white house on a cloudy day.*

The user pivots one axis to the "weather" hierarchy and begins to drill down into tags which might relate to a cloudy day such as "overcast" and "cloudy".

> *A red car beside a white house on a cloudy day. I had driven for over an hour to get here.*

With the knowledge that the lifelogger had driven for an extended period before the target image, the user begins to explore the contents of cells in the projection space which look promising. When they locate an image which matches the task description, they use the timeline function to move back in time and investigate if the lifelogger was driving prior to the image being captured.

> *A red car beside a white house on a cloudy day. I had driven for over an hour to get here. It was a Saturday.*

To filter out all results which do not occur on a Saturday, the user simply needs to use the dimension explorer to apply the "Saturday" tag as a direct filter. This will re-populate the projection space so the user can return to investigating the now reduced amount of results.

> *A red car beside a white house on a cloudy day. I had driven for over an hour to get here. It was a Saturday in August.*

Similar to filtering a specific day, the user can very easily apply "August" as a direct filter, refining the results in the projection space even further.

> *A red car beside a white house on a cloudy day. I had driven for over an hour to get here. It was a Saturday in August and it was in the early afternoon.*

It is likely that the user will have been able to submit at least one or two potential results by this stage, but if they are still in doubt, they can further reduce the results in the projection space by filtering out specific hours of the afternoon.

## 4 CONCLUSION

The ViRMA system prototype is the first iteration of a novel virtual reality multimedia analysis platform based on the $M^3$ model. In this paper we have attempted to describe the system with respect to its underlying data model. It is our hope that we can evaluate our approach via the LSC to serve as a benchmark against other multimedia analytics systems.

**Acknowledgement:** This work was supported by MCSA-IF grant 893914.

## REFERENCES

[1] Aaron Duane, Björn Þór Jónsson, and Cathal Gurrin. 2020. VRLE: Lifelog Interaction Prototype in Virtual Reality: Lifelog Search Challenge at ACM ICMR 2020. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge* (Dublin, Ireland) *(LSC '20)*. Association for Computing Machinery, New York, NY, USA, 7–12. https://doi.org/10.1145/3379172.3391716

[2] Snorri Gíslason, Björn Þór Jónsson, and Laurent Amsaleg. 2019. Integration of Exploration and Search: A Case Study of the M 3 Model. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 11295 LNCS. 156–168. https://doi.org/10.1007/978-3-030-05710-7_13

[3] Cathal Gurrin, Björn Þór Jónsson, Klaus Schöffmann, Duc-Tien Dang-Nguyen, Jakub Lokoč, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Graham Healy. 2021. Introduction to the Fourth Annual Lifelog Search Challenge, LSC'21. In *Proc. International Conference on Multimedia Retrieval (ICMR'21)*. ACM, Taipei, Taiwan.

[4] Pascal Mettes, Dennis C Koelma, and Cees GM Snoek. 2016. The ImageNet shuffle: Reorganized pre-training for video event detection. In *Proc. ACM ICMR*. 175–182.

[5] Daniel Seebacher, Johannes Häußler, Manuel Stein, Halldor Janetzko, Tobias Schreck, and Daniel A. Keim. 2017. Visual analytics and similarity search: concepts and challenges for effective retrieval considering users, tasks, and data. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10609 LNCS (2017), 324–332. https://doi.org/10.1007/978-3-319-68474-1_23

[6] Mel Slater and Sylvia Wilbur. 1997. A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and Virtual Environments* 6, 6 (12 1997), 603–616. https://doi.org/10.1162/pres.1997.6.6.603

[7] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. 2017. Aggregated Residual Transformations for Deep Neural Networks. In *Proc. CVPR*.

[8] Jan Zahalka and Marcel Worring. 2014. Towards interactive, intelligent, and integrated multimedia analytics. In *2014 IEEE Conference on Visual Analytics Science and Technology, VAST 2014 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., 3–12. https://doi.org/10.1109/VAST.2014.7042476