Large-Scale MMBD Management and Retrieval

Manish Devgan and Deepak Kumar Sharma

Abstract This chapter explores the field of Multimedia Big Data management and retrieval. Multimedia data is a major contributor to the big data bubble. Therefore, we require separate databases for storing and managing it, hence, the chapter covers all the requirements of a Multimedia DBMS. Multimedia data modelling has also been covered since multimedia data is mostly unstructured. Further, the chapter covers the annotation and indexing techniques that help manage the large amount of multimedia data and finally followed by a detailed description about different databases that can be used for storing, managing and retrieving the Multimedia Big Data. Different databases such as SQL and No-SQL approaches are discussed such as Graph, Key-Value DBs, Column Family, Spatio-temporal Databases.

Keywords Big data · Database management · Storing multimedia data · Indexing of MMBD · Performance and retrieval capacities · Different databases · Graph DBs · Spatio-temporal · Data modelling

1 Introduction

In the recent years, with the emergence of mobile and internet technologies, the world has seen a massive growth in the use of multimedia such as video, images, text and audio, etc., and this has resulted in a big revolution in multimedia data management systems (Wikipedia.com 2018). With the emergence of new technologies and the advanced capabilities of smartphones, smart televisions and tablets, people, especially younger generations, spend a lot of time on the internet and social networks to communicate with others to share information [1]. This information can be in the

M. Devgan · D. K. Sharma (✉)
Division of Information Technology, Netaji Subhas University of Technology, (Formerly Known as NSIT), New Delhi, India
e-mail: dk.sharma1982@yahoo.com

M. Devgan
e-mail: manish.nsit8@gmail.com

S. Tanwar et al. (eds.), *Multimedia Big Data Computing for IoT Applications*, Intelligent Systems Reference Library 163,
https://doi.org/10.1007/978-981-13-8759-3_9

form of text, audio, image or even video graphics. This vast amount of information is called 'big data'.

Unlike traditional alphanumeric data, multimedia data is usually unstructured and noisy. Conventional data analysis is not a feasible mode to handle this huge amount of complex data. Therefore, more comprehensive and sophisticated solutions are required to manage such large and unstructured multimedia data.

The biggest challenge of big data analytics is how to reduce the time consumed to store and manage this data while producing accurate results from the datasets. Multimedia big data explains what is happening in the world, emphasizes hot daily news, shows special events and can be used to predict people's behaviour and preferences.

In this book chapter, we first introduce the basics of Multimedia data and the emergence of Big Data in Multimedia. We discuss the requirements that are essential for a Multimedia Database Management System to function properly and produce efficient results. Further, the chapter covers the annotation and indexing techniques that help manage a large amount of multimedia data. Finally, a detailed description of the databases that can be put to use for storing, managing and retrieving the Multimedia Big Data. The aim of the chapter is not just to make the reader understand the concept of managing the data but also to allow them to question the possibilities of improving the already available methods.

1.1 What Comprises Multimedia Data

'In computation, data is *information* that has been translated into a form that is efficient for movement or processing, Relative to today's computer systems and transmission media, data is information that has been converted into digital form'. It is acceptable for data to be used as a singular subject or plural subject. Raw data is the term that is mostly used to designate data in its most primitive form. Data is, as of today, one of the most important factors that define the company's value. The better data collection and processing is applied the better is the outcome of the product or project.

Terms like '*data processing*' and '*electronic data processing*' have made data a big influence in the field of business computing, which for a time, came to encompass the full gamut of what is now known as information technology. With the advent of computational statistics in corporate world, a distinct data profession termed as corporate data processing emerged.

Currently, we no longer just communicate using text as the only source of communication. There are multiple media that we can use to transmit the information. Multimedia data is a term that is composed of 'Multimedia' and 'Data', Multimedia is further a split of 'Multiple Media', that can be used for transmission and Data is the statistic collected that will be processed and analysed. Multimedia data can comprise of any one or more of the following media such as text, audio, still images, animation, video or other types of media.

Multimedia data can be stored, easily recorded, displayed and even interacted with or accessed by multimedia processing devices, and can also be a part of live performance. Currently, most of the electronic devices are capable of all multimedia functionalities. Multimedia is refining the way we share and interact with data. In contrary to how we share them, it also affects our decisions since products and companies use data analytics on multimedia data to predict future and provide a better solution to our queries.

1.2 Big Data in Multimedia

Data is, as stated above, information that has been translated into a form that is efficient for movement or processing. With the advent of mobile technology and Internet, there has been a rapid increase in the amount of multimedia data that humans have produced in a very few times. Today, we generate over 2.5 quintillion bytes of data everyday which consists of text, audio, video and other types of multimedia data, which is shared on data sharing and social media platforms such as Facebook, Instagram and more (IBM, 2013). The increase in the use of Internet of Things (IOT) products are also contributing to the increasing rate of data being produced and dumped over the internet every single moment.

According to recent studies and surveys, around 527,760 images are shared on Snapchat every minute of the day, which is an image sharing platform service popular nowadays [2]. More than 120 people register every minute on LinkedIn, people watch over 4M YouTube videos every minute and post around 50k images on Instagram, an online social media platform (Koetsier, n.d.)

The above facts are enough to develop an understanding of the amount of data being dumped onto the server farms every day. Most of this data is image and video data and rarely comprise of texts. So, we can easily attach the term ‘Big’ with Multimedia Data. Big data is often characterized by the Five V’s, which are Velocity, Volume, Value, Variety, and Veracity. These characteristics make the big data different from ‘data’ (Fig. 1).

In the next section, we will study about the requirements that a Multimedia Database Management System must have.

2 Requirements of a MMDBMS

There exist not many differences in the requirements for a Multimedia Database Management System (MMDBMS) than a regular Database Management System (DBMS). MMDBMS works alike a DBMS that is used for storing, accessing and managing the data.

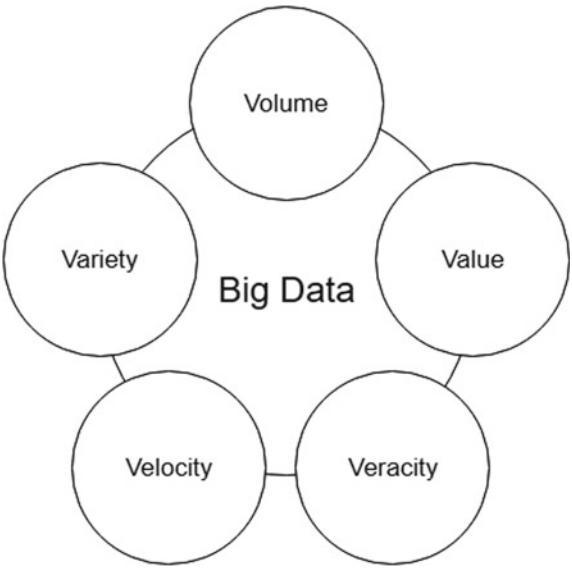

Fig. 1 Five V's of big data

There are several properties that serve as the basic requirements of a DBMS but additional capabilities such as the managing huge data, query support and intractability with multimedia data is defined as MMDBMS. Let us define the traditional capabilities in the section below.

2.1 Traditional Capabilities

Any MMDBMS should be able to serve as a regular DBMS before it can cater to the needs of multimedia data. It must have the basic capabilities as displayed by software such as *Oracle (Oracle.com, 2018), FoxPro (VisualFoxPro, 2018), SQL Server,* etc. A list of basic DBMS capabilities is listed below.

- Providing Data Definition Capabilities
  - Defining a DDL and providing a User-Accessible catalogue
- Providing facilities for storing, retrieval and updating data
  - Defining a Data Manipulation Language
- Supporting multiple views of data
  - An interacting application must see only the required information
- Providing facilities for specifying integrity constraints

  - General Constraints
  - Primary key constraints
  - Secondary Key Constraints

- Controlling access to data
  - Preventing unauthorized access to the stored data
- Concurrency Control
- Supporting Transactions
- Database recovery and maintenance
  - Bringing data back to the consistent state in case of system failure
  - Unloading, reloading, validation etc.

The features defined above are the basic functionalities that a DBMS must possess. A MMDBMS must have other multimedia and big data-specific features as well such as data modelling, huge capacity storage.

2.2 Multimedia Data Modelling

Before we understand the use of data modelling when dealing with multimedia data, we first need to understand what exactly is data modelling? Why is it needed? How is it related to multimedia?

What is Data Modelling?
Data modelling is a software engineering concept of processing raw data to make a data model for any information system using some or the other formal techniques. A data model is a modelling technique, i.e. organizes the elements of data and standardize as to how they relate to each other and to the real world [3].

Need for Data Modelling?
Data modelling techniques and methodologies are used to model data in a consistent, standard, predictable manner in order to manage it as a valuable resource. Any project that requires a standard means of defining and analysing the must follow set defined data modelling standards, which are [3]:

- Assisting programmers, business analysts, manual writers, testers, IT package selectors, engineers, managers, related organizations and clients to understand and use an agreed semi-formal model;
- Managing data as a resource;
- Integrating information systems;
- Designing databases and data warehouses (Fig. 2).

With the advent of IOT and devices such as smartphones, there is a humongous amount of multimedia data that get registered every day on the face of the internet

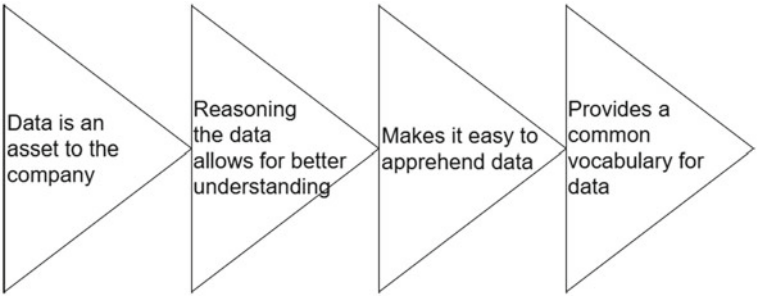

Fig. 2 Importance of data modelling

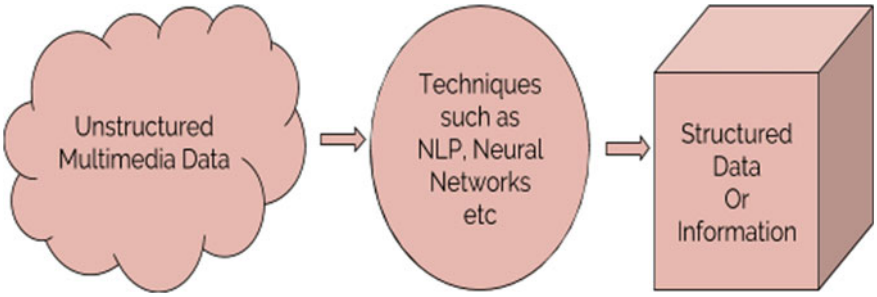

Fig. 3 NLP techniques for modelling data

and cloud storage. Managing, storing and retrieving this immense amount of data is not just tough but also time-consuming. Using techniques such as data modelling can certainly help in aiding to this problem of managing Multimedia Big Data (MMBD).

Although there are various existing models for traditional databases, such as network, relational and semantic models, only very few have been proposed for multimedia databases. The unique nature of multimedia data requires object-oriented data models for each type of media data that exists [3] (Fig. 3).

The question still is that how can we apply the theory of data models on such a vast amount of unstructured data to manage it properly? The problem is generally solved in a conventional way of applying specific schemes to data to make '*sense out of it*'. In recent times, multimedia has contributed to a vast amount of information being produced over the internet in the last decade. Information retrieval or data modelling from unstructured multimedia data is done using advanced techniques such as Natural Language Processing (NLP), Neural Networks (NN) and more. A combination of above schemes along with Object-Relational Database Management System (ORDBMS) is also used to provide semantic retrieval. The ultimate goal of data modelling is to allow the automatic retrieval of target information based on formal information of the related domain.

2.3 Huge Capacity Storage and Management

Multimedia such as images and videos can vary in size ranging from a few megabytes to 3–4 gigabytes and even more. Due to the high volume and high variety of multimedia big data, the problem arises of storage management, which is characterized by its huge capacity of data storage and hierarchical structure. Appropriate storage mechanism for multimedia must be employed in order to achieve better results with analysis and predictions. Multimedia data comprises of interrelated objects that need to be stored perfectly to increase the throughput of the program (a program here can be anything from a large-scale software to a simple look-up machine for the stored data). These objects are placed in a hierarchical storage that can range from online to offline, with increasing storage capacity and decreasing performance. An example can be a simple server with, let's say '*n*' levels of data storage, the first '*m*' levels may use a Solid-state Storage Device (SSD) for storing the data whereas the rest '*n-m*' levels can have greater storage capacity but less accessibility speed due to the usage of Hard Disk Drive (HDD).

There are a few examples of software that allow easy storage of big data, big multimedia data as well, which are listed below:

- Apache Hadoop [4, 5]

Apache Hadoop is a free and open-source software framework that allows effective storing of data in clusters. It runs parallelly and is capable of performing computation on every node simultaneously. It is a JAVA-based framework and uses the Hadoop Distributed File System (HDFS) as the storage system of Hadoop, which splits big data and distribute across many nodes in a cluster. This also replicates data in a cluster thus providing high availability. (hadoop.apache.org 2018) [6]

- Microsoft HDInsight

HDInsight is a cloud-based wrapper around Apache Hadoop provided by Microsoft.

- NoSQL

SQL can be used to handle structured data very well, but we need NoSQL (Not Only SQL) to handle unstructured data. NoSQL databases are schema-less databases. Every row can have a set of columns different from each other. They provide better performance in storing very large amount of data. There are many open-source NoSQL DBs available to analyse Big Data (Wikipedia & TechTarget 2018):

- Hive,
- Sqoop,
- Polybase, etc.

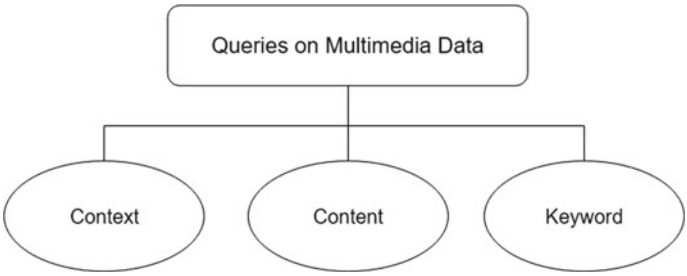

Fig. 4 Queries allowed (keyword, context and content based)

2.4 Query Support and Information Retrieval Capabilities

As we have discussed that the incoming multimedia data is enormous in amount, therefore, there can be numerous queries pertaining to a particular media. This high variety of multimedia data also requires multiple types of query support. It is not necessary to have direct queries produce an accurate result with multimedia data since it is unstructured in nature. A direct query is a query where the exact match for an object is the desired as the result but in case of multimedia data, instead of an exact match, the multimedia query usually results in a list of objects which is closely or somewhat related to query and are ranked in accordance of the relevance they have to the query (Fig. 4).

Therefore, to comprehend data of multiple types, there needs to be a different set of metrics that define the ranking strategies and mechanisms of different data types possible.

2.5 Multimedia Interface and Intractability

The diverse nature of multimedia data requires an intuitive and interactive interface for viewing and interacting with the MMDBMS. Before adding an interactive user interface, there is a need to integrate and compose the data. Data is first broken down into sub-pieces of information with an aim to integrate them into a form such that it can be easily presented.

Data integrity, as well as the uniqueness of multimedia data, must also be preserved. For example, data in the form of pictures and video may consume additional space due to it being distributed in pieces. This is called redundancy of data and leads to reduced memory space for unique data.

Once the composition of data is done, the next requirement becomes an interactive interface. Different media require different interfaces for presentation and query. Demand-based handling and retrieval of multimedia assets must also be a feature of the MMDBMS. It must be able to serve the purpose of indirect queries on multimedia

data which appeals to the user. Although serving indirect queries is an important issue but the performance of the system should not be affected.

2.6 Performance

Considering the five V's of multimedia big data, the performance of the DBMS is a big and important requirement which must be fulfilled in order to ensure the productivity of the product/application. The system must be efficient, reliable, supports real-time execution of queries, must guarantee delivery of multimedia assets in order of query, their integration and the Quality of Service (QoS) [7] acceptable to the user.

Therefore, a DBMS should be fast and secure. Neither should be compromised for the other in order to ensure a perfect database management system for multimedia big data.

3 Annotation and Indexing of Multimedia Big Data

In this chapter, we have been dealing with the topic of creating a DBMS software that allows us to easily store, retrieve and maintain the multimedia data. In any data storage software, annotation and indexing of records play a very important role. It determines the speed and also the accuracy by which the data will be retrieved for a particular query submitted into the MMDBMS.

Let us study about what annotation and indexing are in terms of a DBMS and MMDBMS.

3.1 Multimedia Annotations

'*Annotation* is a note added to a text, for example, a comment beside a topic of the book is an annotation' [8]. In real life, annotation helps in understanding the topics better as well as makes the lookup of different topics and categories a lot easy for the reader of the book.

Similarly, annotations play a significant role in assisting in data management and retrieval, specifically for heterogeneous and unstructured data. The data that is raw and needs a significant amount of cleaning is benefited from this technique [9] (Fig. 5).

There are two main categories of multimedia annotations:

1. Manual,
2. Automatic.

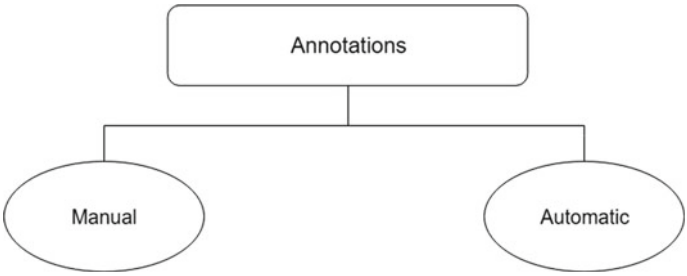

Fig. 5 Manual and automatic annotations

Since the data is to mark with specific notes, it can either be done by the user or a machine can be employed to do the same work. Therefore, two kinds of annotation techniques exist.

Manual Annotation

The manual process of annotation of incoming multimedia data is very time consuming for the uploader/user. The annotation process is about understanding the multimedia data from start to end (in case of audio and video) and marking all annotations in order of occurrence.

Although this process need not give the desired result, therefore a set of rules must be followed by all the annotators while making annotations for multimedia (or any) data.

- Reading/Understanding the data
  - Viewing the data of the multimedia image, video or audio in its entirety to generate an understanding of the raw available data.
- Marking the entities
  - This step involves a proof reading of the document and marking any available entities that occur in the document
- Looking Again
  - Reviewing your work to be certain about the fact that no possible annotation is missed and also that the features are correctly mapped according to their occurrence in the data.
- Recording any additional information
  - The annotator is expected to record the experience while annotating the document/data. This can be done using other tools to make comments or sometimes can be done using the annotation tool itself (Fig. 6).

These sources of annotation are reliable but this process is very time-consuming as it wastes a big amount of time from the user's side to add comments or metadata about

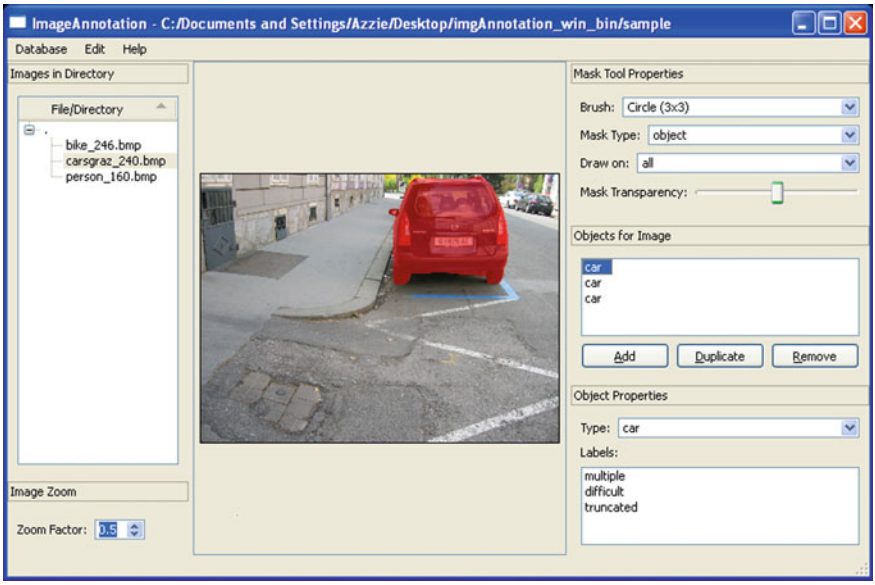

Fig. 6 LEAR is an image annotation tool for windows

the multimedia data. The problem becomes even more serious when the growing entropy of the big data bubble is taken into consideration.

Therefore, a new faster method must be devised to prepare annotations for the multimedia data. So, we have the automatic annotation process.

Automatic Annotation Process

The automatic annotation process is backed by the highly sophisticated machine learning algorithms, which is more appealing than the manual method, considering the ever-increasing amount of data. However, it is more challenging because of the notorious semantic map problem. Semantic mapping is a process of creating a map which can be used *to visually display the meaning-based connections between a word or a phrase and a set of unrelated words*. With access to the shared information available over the connected network and to heterogeneous sources, the problem involving terminology provision and interoperability of systematic vocabulary schemes still exist and require urgent attention [10]. Therefore, solutions are needed to improve the performance of full-text retrieval system.

Although using machine learning algorithms to ease out the work seems a good option for the difficulties associated with its implementation must also be taken care of.

In spite of big challenges and difficulties of multimedia automatic annotation, there have been many research endeavours in this hot topic. One such research is to use Latent Dirichlet Modelling (LDA) to extract semantic topics from multimedia

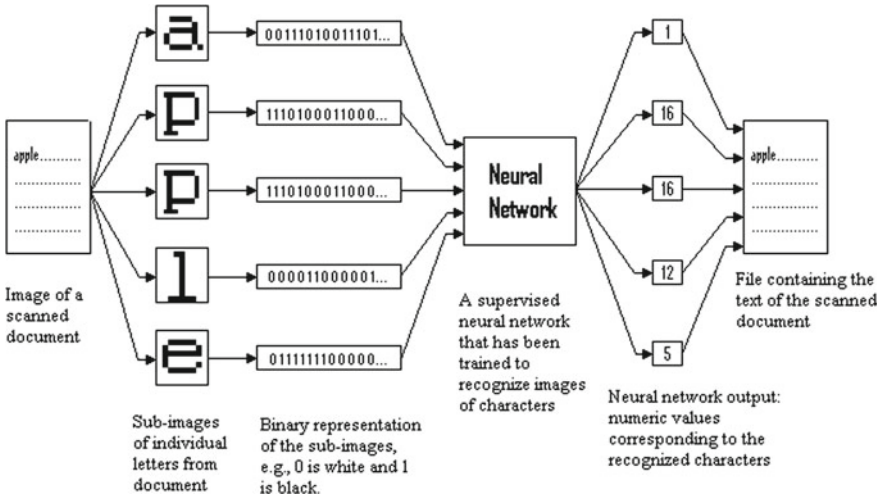

Fig. 7 Using neural nets to identify text from images to provide better annotations

text documents such as still images [11]. Some other seek the combination of both humans as well as the computer for multimedia data annotation.

Deep Learning, a field of Machine Learning, involving the extensive use of Artificial Neural Networks is also being used to generate annotations for images, videos, and audio. Deep Nets have shown promising results when compared to conventional machine learning algorithms.

Figure 7 shows how we can use images of scanned images and convert it into a file of the same using supervised neural networks. It is a case of Optical Character Recognition or OCR, which is used to convert written text into digital text by identifying the words in an image using a pre-trained neural network.

3.2 Multimedia Indexing

As mentioned earlier, multimedia big data is mostly unstructured. This means that the data is lost in a relational sense. While the traditional RDBMS were used for managing the structured data, this means that they could not be used for managing multimedia data without any change. To solve this problem, a number of indexing strategies have been proposed, targeting different types of data and queries. The indexing strategies can be roughly categorized into AI approaches and the non-AI approaches [12].

Specifically, the non-AI approaches treat each individual item independently and do not rely on the context or data relationships. They can be further classified into tree-based strategies, such as *R-Tree, B-Tree, Hash, X-Tree, Gist* and inverted indexing cloud services can be answered by retrieving and ranking the corresponding list of documents [13, 14]. On the other hand, the AI approaches capture the data semantics by analysing the relationship between the data and the context among them to better structure and organize the multimedia data items. These advantages of AI approaches make them more accurate and effective than the non-AI methods. However, they are generally more complicated and computationally expensive. The AI methods are still a field of vast studies and research [15].

4 Storage and Retrieval

As discussed before, we know that storage is the main purpose of why we employ databases in the first place. It is to store the important information and use it thereafter, for example, to enhance the customer experience. There are multiple types of databases that can be employed to store the multimedia data, and each of it has its unique feature (Fig. 8).

There are two basic types of databases that can be employed to store the multimedia data so that it can be used further, viz. SQL and no-SQL. Their capabilities of storing and retrieving vary to different extents, so we are going to study about them.

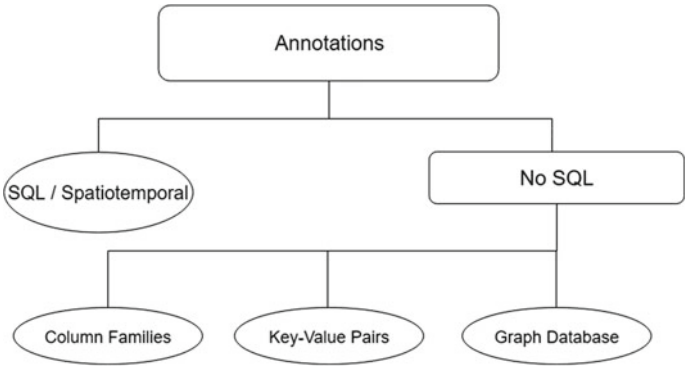

Fig. 8 Spatio-temporal and NoSQL databases

4.1 Spatio-temporal Databases

Spatio-temporal means to consist of both space and time. In a spatio-temporal database, as the name suggests, the emphasis is laid on space-time information.

Use case of Spatio-temporal Databases
Spatio-temporal database is capable of analysing data that exists in either space, time or both of the frames. It is therefore suitable for the following purposes:

- Keeping track of moving objects. The objects in a moving plane are known to exist at a single position at any given frame of time. This is an apt example of the space-time dependent data.
- A data consisting of wireless communication networks, which may exist only for a short timespan within a particular geographic region
- Maintaining an index of animal species in a given region, a new species may be found or an old one can become extinct, so this is also an example for spatio-temporal data
- Tracking historical plate data and more

We can also make use of artificial intelligence to predict the behaviour of the object that has the characteristics stored in a spatio-temporal database.

Since multimedia is unstructured raw data, therefore, using spatio-temporal databases for time-dependent data is a big advantage for managing and storing time-dependent data. Some example of spatio-temporal data areGPS, payments, digital photos, any smartphone data, etc. (Figs. 9 and 10).

Implementing Spatio-temporal DB
Since there are no RDBMS that incorporate a spatio-temporal database therefore it must always be implemented so that it can be used. Since the extensions do not exist, so it is very difficult to implement it. Software such as TerraLib (TerraLib, Open Source), which is an open-source project, uses a middleware approach for storing data in a relational database. The theory of this area is not completely developed. Another approach for this is to use constraint database system such as MLPQ or Management of Linear Programming Queries (Fig. 11).

4.2 No-SQL Databases

A NoSQL database provides a software mechanism for storing, managing and retrieving data that is modelled in means other than the relationaltables used in relational databases [3].

Big data and real-time web applications make consistent use of NoSQL Databases.

Many NoSQL stores compromise on consistency in favour of availability, partition tolerance and speed. It is not used to a much greater extent due to its use of low-level query languages, lack of standardized interfaces, and huge previous investments in

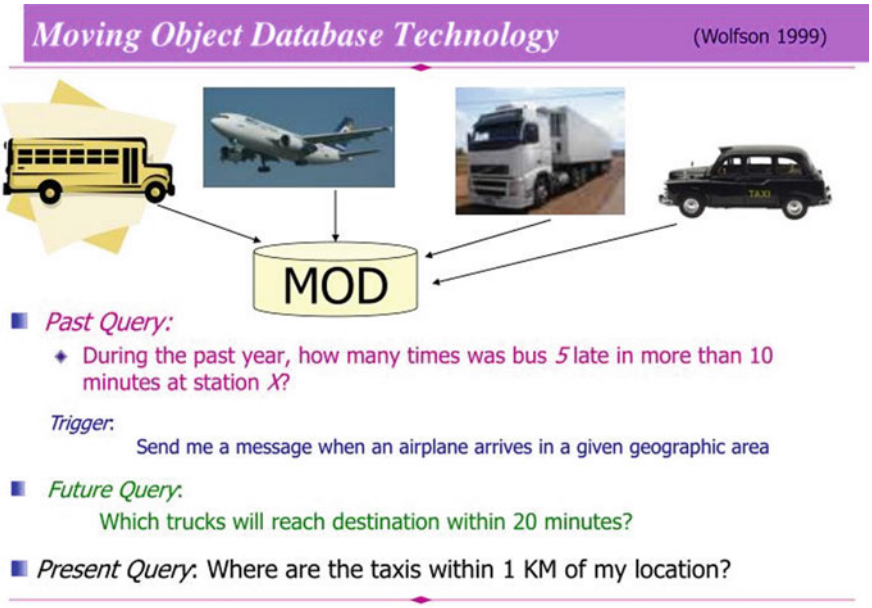

Fig. 9 Examples of using spatio-temporal queries

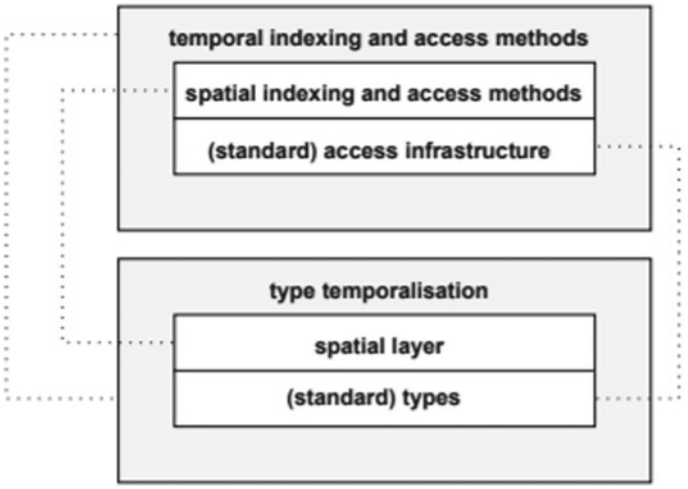

Fig. 10 The storage manager in a spatio-temporal DBMS

Fig. 11 TerraLib and Geomesa are open-source tools

existing relational databases. But it's ability to be able to hold multiple types of data under a single collection is what makes it so popular and a good choice for storing multimedia data.

NoSQL Databases—Types

There are multiple types of NoSQL Databases that are used to store, retrieve and manage data. Some of these may overlap with the other in terms of functionality. Below is a classification of NoSQL tables with respect to data model:

- Column,
- Document,
- Key-Value,
- Graph.

Uses of NoSQL DBs

NoSQL is not specific to store just text but all kind of data, be it large or small, text or image, it works perfectly on any kind of data. It is considered a distributed, efficient, and non-relational database. Most of the conventional database cannot store and manage the Binary Large Objects data or BLOB data. BLOBs are a set of default binary data that can be used to store any type of multimedia objects. However, unstructured multimedia data can be stored, processed and analysed using NoSQL schema-less documented-oriented architecture to treat data more efficiently and flexibly [16].

4.2.1 Key-Value Stores

Key-value databases are designed to *store, retrieve and manage* a collection of objects which can be string, JSON, BLOB or anything [3]. It is a giant mapping of a key with its associated value. This collection of objects is generally called as a dictionary or a hash. Each object is a record in a dictionary is identified by a unique key that is used to quickly retrieve the data in the database.

Compared to relational databases, key-value store significantly improves the performance due to the cache techniques used for mapping. Amazon Simple Service S3 (Dynamo) is a cloud-based key-value store that is most commonly used for storing large data (Fig. 12).

Key-Value databases can use *consistency models* which can range from eventual consistency to *serializability* [3]. There are some stores which support ordering

Key	Values
Key 1	AAA, BBB, CCDA
Key 2	2018/09/01, BAAD
Key 3	3,ZAA, AABD
Key 4	AAB, CCD, KFC

Fig. 12 Key-value representation of data

of the keys. Some of the stores manage the data inside the memory, RAM while others simply employ secondary storage such as solid-state drives and/or hard disk comprising of the rotating disks.

Redis is on one of the most popular key-value database implementations. There are other services such as the Oracle NoSQL DB in which every entity is a set of key-value pairs. Every key has multiple components, specified as an ordered list.

The *major key* is used to identify the entity and generally consists of the *leading components* of the major key [3]. This means that one can move from the outer key to inner key easily. This is similar to the way in which the directories are arranged. One directory inside the another. Whereas can be any string of object, text or even image encoded as text. Some examples of Key-Value Store/Database are Apache Ignite, Redis, Level DB and Oracle NoSQL Database (Fig. 13).

Amazon S3 is a cloud bucket which has key-value service and is utilized as cloud storage service because of its encryption and security mechanisms. It is simple to integrate and use [17].

One of the problems of the key-value store is the fact that key-value technique lacks consistency or atomicity of the transactions. In addition, maintaining the keys for a very large-scale data may be troublesome for use in general as well as may result in a greater time complexity for search and retrieval.

4.2.2 Graph Database

So far, we have discussed SQL and NoSQL databases, now we will study about graph databases. Graph database as the name suggests is a database that makes extensive use of graph as a sole data structure associated with this database.

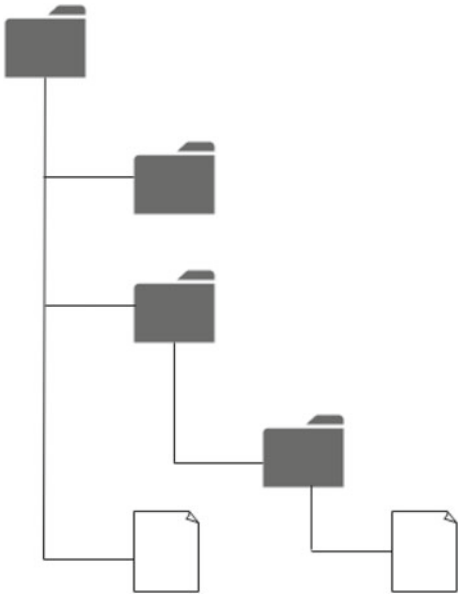

Fig. 13 An example of directory listing

Design of a Graph DB

The key concept of this design is the node and edge system of the graph, which allows the data to be linked to any node in any possible way. This allows a multi-relational design in which any data can be directly or indirectly connected to other using edges. This is different from the relational database design as the links in the relational tables are logical and can be consumed with operations such as 'join' but in case of graph database they can simply be done using the physical links between the nodes. Executing relational queries is possible with the help of the database management systems at the physical level [3], which allows boosting the performance without modifying the internal logical structure of the database. The main advantage of the graph structure is *the simple and fast retrieval of complex hierarchical structures* that are difficult to model and implement.

Normally, a graph database is a NoSQL structure since it uses key-value structure or document-oriented database for storing data (Fig. 14)

Properties of Graph DB

One of the important properties of the graph database is the use of tags or properties to annotate the stored. Tags are just another pointer to a document of a similar kind. This allows for easy mass retrieval of data at once.

Like other NoSQL databases, maintaining the consistency of data is not inherently done by the graph databases, and hence, the application system is responsible for governing the consistency over the graph data.

The other tough tasks that are included in this data structure is to scale out the graph database and finding a way to design generic queries.

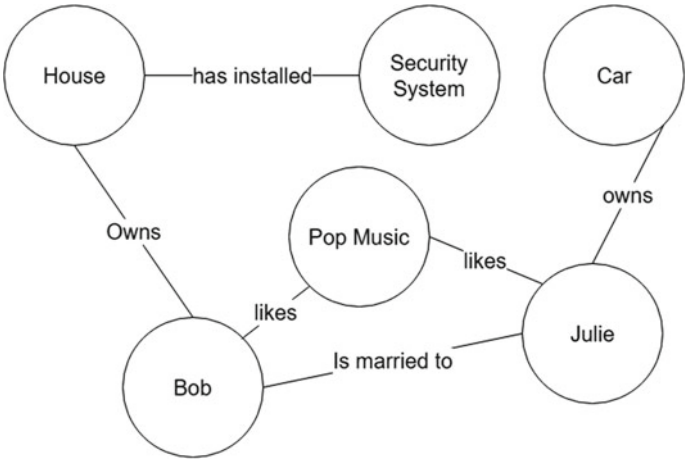

Fig. 14 Graph data

Data retrieval from a graph database requires a different query language that is not just simple SQL, but it needs to be better than SQL and can therefore support the graph structure. There have been many graph query languages but none has been accepted as the industry standard as SQL was for relational databases. Most graph databases allow access using an application programming interfaces or APIs.

Example usage of graph databases is to perform queries like '*computing the shortest path between two points in a graph*'. Graph also makes different other queries easily possible.

4.2.3 Column Families and Document Stores

Unlike relational DBMSs (RDBMSs) that store data in rows, column-based NoSQL databases store data in columns, which results in a fast and scalable data search. In column family databases, all the cells related to a column are placed in a continuous disk entry while in an RDBMS, each row is stored in different parts of the disk. Therefore, wide-column stores allow extremely large and unstructured data management, which is essential in multimedia big data analytics. They can also scale well in very big datasets where non-complex key-value pairs are insufficient. Google's BigTable, Cassendra and HBase are the most popular column families. BigTable [18] for instance, has several main advantages, such as data sparsity, distribution across multiple clusters, multidimensionality, persistence and key-value mapping.

However, one of the major downsides to column families is the expensive writes as data are split into several columns, while this cost can be amortized in very large streaming writes. In document-based NoSQL databases, data including a set of key-value pairs are stored as documents where the values have an encoded structure such as XML and JSON.

Advantages
The advantage of this structure is that the document stores embed metadata aliased with the data contents. In addition, any complex data can be represented as a document and can be stored in these databases. However, this powerful schema-less characteristic can also raise the potential for accidents and abuse, as well as performance issue.

MongoDB—Popular Open-Source DB
MongoDB is a popular document-based database that is publicly available and is written in C++ (MongoDB.Com 2016). It usually stores and groups documents based on their structure. In MongoDB, Binary JSON is the encoded format used to store unstructured or semi-structured documents such as multimedia and social media posts, and the indexing is very similar to that of relational databases. MongoDB has several advantages, such as durability using the Master–Slave replication technique and concurrency using locks, while the main limitation of this database is its limited data storage per node. It is also widely used for social data and multimedia big data analytics.

5 Conclusion

In this chapter, we've studied about what is multimedia data, how can it be stored and what are the ways in which the database can be made ensuring that this unstructured data can be stored inside it. We have also learned about the typical characteristics about the multimedia big database management system. The things that define a good database system are the traditional database capabilities, data management and huge capacity storage, multimedia data modelling, media integration and a lot more.

We've also studied about how annotation can help in making the data retrieval process easy and the ways to annotate the multimedia or any regular data. The process of indexing can be a much helpful process when it comes to gather information and execute queries.

Finally, we studied about various types of databases that can be used to store the multimedia big data. The only two types are SQL and NoSQL, for example, the spatio-temporal database that deals with both space and time and NoSQL are divided into further types such as the key-value, document store, graph database, etc. We read about how different techniques work, how they are implemented and how can we use them in regular day to day for a company or a project which has multimedia data at a large extent.

There were citations of different software that can be chosen as an apt database for the use case as and when desired.

The purpose of the chapter was to impart knowledge about the field of database management and retrieval in multimedia big data, readers can also check out the references to other research papers and websites to read more.

References

1. S.-C. Chen, Multimedia databases and data management: a survey. Int. J. Multimedia Data Eng. Manage. 1(1), 1–11
2. E. Adler, Social media engagement: the surprising facts about how much time people spend on the major social networks (2016). Retrieved from http://www.businessinsider.com/social-media-engagement-statistics-2013-12
3. V. Abramova, J. Bernardino, NoSQL databases: MongoDB vs Cassandra, in *Proceedings of the International C* Conference on Computer Science and Software Engineering*. (ACM, 2013), pp 14–22
4. Hadoop, Apache Hadoop (2018). http://hadoop.apache.org
5. Mahout, Apache Mahout (2018). http://mahout.apache.org
6. D.A. Adjeroh, K.C. Nwosu, Multimedia database management—requirements and issues. IEEE MultiMedia **4**(3), 24–33 (1997)
7. Kumari, S. Tanwar, S. Tyagi, N. Kumar, M. Maasberg, K.K.R. Choo, Multimedia big data computing and internet of things applications: a taxonomy and process model. J. Netw. Comput. Appl. **124**, 169–195 (2018)
8. V. Alvarez, S. Richter, X. Chen, J. Dittrich, A comparison of adaptive radix trees and hash tables, in *Proceedings of the 31st IEEE International Conference on Data Engineering*. 1227–1238 (2015)
9. F. Amato, F. Colace, L. Greco, V. Moscato, A. Picariello, Semantic processing of multimedia data for e-government applications. J. Vis. Lang. Comput. **32**(2016), 35–41 (2016)
10. S.-C. Chen, R.L. Kashyap, A spatio-temporal semantic model for multimedia database systems and multimedia information systems. IEEE Trans. Knowl. Data Eng. **13**(4), 607–622 (2001)
11. P.K. Atrey, M. Anwar Hossain, A. El Saddik, M.S. Kankanhalli, Multimodal fusion for multimedia analysis: a survey. Multimed. Syst. **16**(6), 345–379
12. F. BintaAdamu, A. Habbal, S. Hassan, R.L. Cottrell, B. White, I. Abdullahi, A Survey on Big Data Indexing Strategies. Technical Report. SLAC National Accelerator Laboratory (2016)
13. D. Che, M. Safran, Z. Peng, From big data to big data mining: challenges, issues, and opportunities, *Database Syst. Adv. Appl.* (Springer, Wuhan, China, 2013), pp. 1–15
14. K. Chatterjee, S.-C. Chen, HAH-tree: towards a multidimensional index structure supporting different video modelling approaches in a video database management system. Int. J. Inf. Decis. Sci. **2**(2), 188–207
15. R. Bryant, R.H. Katz, E.D. Lazowska, Big-data computing: creating revolutionary breakthroughs in commerce, science and society (2008). Retrieved from https://pdfs.semanticscholar.org/65a8/b00f712d5c230bf0de6b9bd13923d20078.pdf
16. T. Chardonnens, Big data analytics on high velocity streams: specic use cases with storm. Master's thesis. Software Engineering Group, Department of Informatics, University of Fribourg, Switzerland (2013)
17. Amazon AWS official docs
18. F. Chang, J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, T. Chandra, A. Fikes, R.E. Gruber, Bigtable: a distributed storage system for structured data. ACM Trans. Comput. Syst. **26**(2), 4:1–4:26