Efficient and robust large medical image retrieval in mobile cloud computing environment

Yi Zhuang a,⇑, Nan Jiang b, Zhiang Wu c, Qing Li d, Dickson K.W. Chiu e, Hua Hu f

a College of Computer & Information Engineering, Zhejiang Gongshang University, Hangzhou, PR China
b Hangzhou First People’s Hospital, Hangzhou, PR China
c Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, PR China
d Department of Computer Science, City University of Hong Kong, Hong Kong Special Administrative Region
e Faculty of Education, The University of Hong Kong, Hong Kong Special Administrative Region
f School of Computer, Hangzhou Dianzi University, Hangzhou, PR China

A R T I C L E  I N F O

Article history:
Received 5 April 2013
Received in revised form 16 September 2013
Accepted 13 October 2013
Available online 21 October 2013

Keywords:
Medical image
Multi-resolution
Mobile cloud

A B S T R A C T

This paper presents an efficient and robust content-based large medical image retrieval method in mobile Cloud computing environment, called the MIRC. The whole query process of the MIRC is composed of three steps. First, when a clinical user submits a query image $I_q$, a parallel image set reduction process is conducted at a master node. Then the candidate images are transferred to the slave nodes for a refinement process to obtain the answer set. The answer set is finally transferred to the query node. The proposed method including an priority-based robust image block transmission scheme is specifically designed for solving the instability and the heterogeneity of the mobile cloud environment, and an index-support image set reduction algorithm is introduced for reducing the data transfer cost involved. We also propose a content-aware and bandwidth-conscious multi-resolution-based image data replica selection method and a correlated data caching algorithm to further improve the query performance. The experimental results show that the performance of our approach is both efficient and effective, minimizing the response time by decreasing the network transfer cost while increasing the parallelism of I/O and CPU.

1. Introduction

In recent years, with the explosive increase of medical multimedia data in hospital information management systems (HIMS), many applications (e.g., health information retrieval [40], clustering [5,17] and recommendation [39], etc.) require a highly efficient access method to support content-based multimedia retrieval at a large scale. As one of the most important media types, medical images and their management, query, and analysis play a critical role in the modern HIMSs. Millions of medical images generated each year present an enormous challenge to healthcare organizations, as they need to efficiently manage, access, and share such data while trying to reduce costs. How significant is this issue? The following are some eye-opening statistics:

- Medical image archives are increasing by 20–40 percent each year. It is estimated that 1 billion medical images are stored in the US by the end of 2012.
- It is estimated that medical imaging information storage constitutes one-third of the global storage demand, which was the equivalent of 1.2 billion average hard drives in 2007.

⇑ Corresponding author. Address: Apart.502, Unit 2, Blg.7, Qinchunfang, Hangzhou, PR China. Tel./fax: +86 571 87247080.
E-mail address: zy158cn@gmail.com (Y. Zhuang).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.10.013
Although considerable amount of research efforts have been carried out on medical image indexing and similarity query processing in high-dimensional spaces \cite{20}, most of these focus on a centralized way (i.e., single-PC-based), which cannot scale up well to large data volume. The query efficiencies of these centralized methods are unsatisfactory because the response time is linearly increasing with the size of the searched file. Therefore the design and development of high performance medical image query methods becomes a critically important research topic.

Cloud services (also known as “the cloud”) refers to a network of servers connected by the Internet (or other networks) that enables users to combine and use computing power on an as-needed basis. Individual user does not have to purchase and maintain his own computing power. The cloud provides virtual centralization of applications, storage, etc., which can be accessed by any web-friendly device (such as computer, laptop, smart phone, tablet) virtually anywhere.

A mobile cloud (MC) can be regarded as a type of flexible computing infrastructure consisting of many computing nodes, which can provide resizable computing capacities to different users anywhere anytime. To fully harness the MC power, efficient data management is needed to handle huge volumes of medical image data and support a large number of concurrent end users (e.g., physicians). In addition, the MC environment provides us with location-based query support that enables clinical users to retrieve patient records and images conveniently. To achieve this, scalable, high-throughput, location-based querying, and indexing schemes are generally required. However, as shown in Fig. 1, for MC-based medical image query processing, exploring parallelism in the MC to speed up the queries is a new research topic, which has received little attention so far. The challenges include three main aspects:

1. **High computation cost in medical image retrieval**: most of medical images are characterized by high pixel resolution, high-dimensional, and large-scale. So the query cost of such medical images is very high.

2. **Mobility of MC users**: most clinical users in the MC are constantly moving. That means the spatial position of each user varies with the variance of time. So, how to perform an optimal data placement is also a challenging issue.

3. **Instability and heterogeneity of the MC**: the nodes in the MC are instable, that means, some nodes may be down or connected intermittently to the network. The bandwidth between any two nodes in the MC may be different according to the variance of time. There is no guarantee that the total response time of each query will be similar.

To address the above challenges, we propose an efficient distributed similarity query processing technique (\textit{MIRC}) in the mobile cloud environment. In particular, the \textit{MIRC} includes four enabling techniques, namely, learning-based optimal data placement scheme, content-aware and bandwidth-conscious multi-resolution-based image data replica selection scheme, priority-based image block data robust transmission algorithm and learning-based dynamic correlated data caching scheme. We have implemented the \textit{MIRC} method and extensive experiments indicate that our approach is specifically suitable for the large high-dimensional image data queries. Without loss of generality, Euclidean Distance is used as the underlying distance function in our approach. The contributions of this paper are summarized as follows:

- We introduce a framework of a efficient and robust large medical image retrieval in the mobile cloud environment (\textit{MIRC}) to improve search efficiency, especially for large-scale high-dimensional image repositories.
- We design a \textit{learning-based adaptive data placement scheme} to maximize the query parallelism at the master node level, in which clinical users’ moving trajectories among different departments are sampled and analyzed to estimate optimal placement positions of the medical image data among different departments.
- We propose a \textit{content-aware and bandwidth-conscious multi-resolution-based image data replica selection scheme} to adaptively perform the data transmission process in a reasonable time.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{mobile-cloud-architecture.png}
\caption{The general architecture of a mobile cloud environment.}
\end{figure}
We present a priority-based image block data robust transmission method to progressively reduce the communication cost in the MC environment and improve users’ experiences.

We devise a learning-based dynamic correlated data caching scheme to improve the efficiency of the query process.

We conduct extensive experimental studies to evaluate the effectiveness, efficiency, scalability, and robustness of our proposed algorithms.

The rest of the article is organized as follows. Section 2 reviews related works and Section 3 presents preliminary definitions. Section 4 introduces four enabling techniques to facilitate a fast similar search over mobile clouds. In Section 5, we propose two indexing schemes, viz., GQ (global query routing index), and MAM (major and minor index). After that, we propose a Minc query processing algorithm in Section 6. In Section 7, we perform comprehensive experiments to evaluate the efficiency of our proposed approach before we conclude the paper in Section 8.

2. Background

In this section, we first provide background review of medical image retrieval and image data transmission techniques. Then we review two algorithms (i.e., the iDistance high-dimensional indexing method and the affinity-propagation (AP) – based clustering algorithm), which will be used in our system.

2.1. (Distributed) medical image retrieval

The image retrieval has been extensively studied for about 40 years so far. Many prototype systems have been developed. Among them, the QBIC system [16], MARS project [30], VisualSEEK system [33] focus on image retrieval, and WebSEEK [34] system is a Web-oriented search engine that can retrieve both images and video clips. These approaches differ from each other in either the low-level features extracted from the data, or the distance functions used for similarity calculation. Despite the differences, all of them are similar in the following fundamental aspects: (1) they all rely on low-level features; (2) they all use the query-by-example paradigm.

In the domain of medical imaging informatics, a huge amount of image data is being produced. A lot of work has already been done to improve the image retrieval systems. There are three ways that medical images are retrieved: (1) text-based method; (2) content-based method and (3) text and content-based method [20]. In text-based image retrieval, the images are retrieved by the manually annotated text descriptions and traditional database techniques to manage images are used. In content-based image retrieval, the images are retrieved on the basis of features such as color, texture, shape, and so on, which were extracted from the images themselves. So far, a variety of medical image retrieval systems have been developed using either method (text-based or content-based) or combining two methods;

2.1.1. Text-based medical image retrieval

Text-based image retrieval system can be traced back to 1970s. Text-based image retrieval system is prevalent in the search on the internet web browsers. Although text-based methods are fast and reliable when images are well annotated, they cannot search in unannotated image databases. Moreover, text-based image retrieval has the following additional drawbacks, it requires time-consuming annotation procedures and the annotation is subjective [20]. Text-based query commonly results in irrelevant images. Thus, to support effective image searching, retrieval methods based on the image content were developed.

2.1.2. Content-based medical image retrieval

Recent trials for content-based medical image retrieval were ASSERT system [32] for high resolution computed tomography (CT) images of the lung and image retrieval for medical applications (IRMA) system [12] for the classification of images into anatomical areas, modalities and viewpoints. Flexible image retrieval engine (FIRE) system handles different kinds of medical data as well as non-medical data like photographic databases [14].

In ASSERT system, the system lets the user extract pathology-bearing regions in lung images and these regions are then characterized by their grayscale, texture, and shape attributes, which are used for image retrieval.

The IRMA system is a generic web-based X-ray retrieval system. It allows the user to extract images from a database given an X-ray image query. Local features and similarity measures are used to compute the nearest images. This system also developed a classification code for medical images to classify medical images based on four axes (modality, body orientation, body region, and biological system) [27]. The local features are derived from the previously classified and registered images that have been segmented automatically. IRMA analyzes content of medical images using a six-layer information model: raw data, registered data, feature, scheme, object, and knowledge. IRMA lacks the ability for finding particular pathology that may be localized in particular regions within the image.

In FIRE system, query by example image is implemented using a large variety of different image features that can be combined and weighted individually and relevance feedback can be used to refine the result [14].

2.1.3. Combined text and content-based medical image retrieval

Considering the intrinsic difference between the text and image in representing and expressing information, there have been approaches to combine the text-based and content-based image retrieval. Techniques that perform the text-based method first [26,3] and two methods at the same time [13,15] were studied.
The hybrid image retrieval systems to incorporate external knowledge that is encoded in lexicons, thesauri and ontologies were suggested [41].

The new project, annotation and image markup (AIM) for medical image annotation and markup is being developed to make all the key semantic content of images machine-readable using controlled terminologies (mainly RadLex) and image markup standards.

The above mentioned systems are based on a single PC environment, their processing scalabilities are limited, especially for a large volume of the medical images. In the research of distributed multimedia retrieval, Berchtold et al. [4] proposed a fast parallel similarity search in multimedia databases by providing a near-optimal distribution of data items among the disks. Furthermore, Papadopoulos et al. [28] presented a similarity search using disk array. Anbarasi et al. [2] presented distributed medical image retrieval method using traditional distributed databases. As one of the most important distributed computing scenarios, Peer-to-Peer (P2P)-based similarity search has been received much attention. Recently, Charisi et al. [10] proposed a content-based medical image retrieval system in peer-to-peer network environment. Although Wang et al. [38] proposed a multi-dimensional index scheme in cloud network, it is not suitable for the mobile environment as characteristics of mobile networks have not been fully considered. So far, fewer research work have touched on medical image query in Mc environment due to the different query mechanism.

2.2. Image transmission techniques

Image data transmission techniques have been studied for about 20 years [11,22,29,1,25,31]. The state-of-the-art methods can be mainly divided into two categories: (1) improvement design of transmission protocol [11,22,29,1,19]; and (2) image data encoding and compression [9,25,31,6,7,8,24,35,37].

Charles and Larry [11] first proposed a wireless image data transmission method from end to end, and provided experimental analysis. As images are usually transmitted across the Internet using a lossless protocol such as TCP/IP, lossless protocols require retransmission of lost packets, which substantially increases transmission time. John et al. [22] presented a fast lossy Internet image transmission scheme (FLIT) for compressed images which eliminates retransmission delays by strategically shielding important portions of the image with redundancy bits. They described a joint source and channel coding algorithm for images which minimizes the expected distortion of transmitted images. After that, Raman et al. [29] proposed an image transmission protocol called ITP. Comparing with the traditional TCP protocol, the ITP is more suitable for image data transmission. Packet losses decrease the quality of image or video in multimedia applications. Robustness to packet losses is crucial to high quality image transmission in Internet or wireless channels where data loss often happens. Gao et al. [19] proposed a robust image transmission scheme for wireless channels based on compressive sensing.

Since most image transmission methods use the same pixel interpolation scheme for the entire picture, without considering the differences in different portions, Chang et al. [9] presented a progressive image transmission (PIT) scheme which transmits the most significant portion of a picture, followed by less important parts. Lin and Hao [25] presented a compound image compression algorithm for real-time applications of computer screen image transmission called Shape Primitive Extraction and Coding (SPEC). Ruiz et al. [31] designed an image compression algorithm to support progressive image transmission. Available PIT mechanisms and systems can be categorized into spatial domain [6], and pyramid-structured progressive transmission [24]. In the transform domain, an image undergoes block compression and the transformed coefficients are transmitted progressively in a relative importance order (e.g., Progressive JPEG). Alternately, a germainal and instinctive method for progressive image transmission in the spatial domain is the Bit Plane Method (BPM) [8,35]. In such scheme, the final transmitted image is the same as the original. However, its high transmission bit rate is a major disadvantage of BPM. Due to the drawback of BPM, lossy PIT techniques have received more attention. In [7], to provide a fast PIT scheme, Chang et al. improved the BPM method by color guessing called the guessing by neighbors (GBN) method which uses interleaved pixels for transmission. Fifty percent of the pixels are transmitted while the other fifty percent were “guessed”.

Different from the above state-of-the-art methods, the paper proposes a content-aware and bandwidth-conscious big medical image data transmission method by analyzing the image content and network bandwidth. To the best of our knowledge, this is the first work to improve the transmission performance from a perspective of data level.

2.3. iDistance

iDistance [21] is a distance-based high-dimensional index method for similarity search. In this paper, we adopt it as an index facility to efficiently support medical image dataset reduction.

In iDistance, data space is first partitioned into $T$ clusters and a reference image $O_j$ for each cluster $C_j$ is selected. Each image is assigned a one-dimensional iDistance value according to the distance to its cluster’s reference image. Having a large constant $c$ to separate individual clusters, the iDistance value for an image $I_i \in C_j$ is as follows:

$$iDis(I_i) = j \times c + d(I_i, O_j)$$

Then, all images in cluster $C_j$ are mapped to the interval $[j \times c, (j + 1) \times c]$. In this way, the problem of similarity search is transformed to an interval search problem. For a range query $\Theta(I_q, r)$, for each cluster $C_j$ that satisfies the inequality $d(O_j, I_q) - r \leq R_j$, the images that are assigned to the cluster $C_j$ and their iDistance values belonging to the interval
\[ j \times c + d(L_q, L_q) - r, j \times c + d(L_q, L_q) + r \] are retrieved. Let \( l_i \) be from these images, the actual distance to the query image is evaluated and if the inequality \( d(l_i, l_q) \leq r \) holds, \( l_i \) is added to the result set thereafter.

2.4. AP clustering algorithm

The affinity-propagation (AP)-based clustering method \cite{5} is an iterative algorithm in which each data point can be viewed as a network node which passes messages to other nodes in order to determine which nodes should be exemplars and which nodes should be associated with those exemplars. An exemplar is the point which best represents other points in its cluster.

To maximize the overall similarity of all data points to their exemplars, the algorithm is based on the ideas of belief-propagation. There are two types of messages sent between data point \( i \) and candidate exemplar \( k \): responsibility \( r(i,k) \) and availability \( a(i,k) \). Responsibility messages are sent from \( i \) to \( k \) and reflect how strongly data point \( i \) favors \( k \) over other candidate exemplars. Availability messages are sent from \( k \) to \( i \) and reflect how available \( i \) is to be assigned to \( k \) currently.

The messages are passed during several iterations in which the evidence accumulates that some points are better exemplars. The algorithm reaches convergence when enough evidence has been formed about exemplars and assignments to exemplars. At this stage node \( i \) is assigned to whichever candidate exemplar \( k \) maximizes the value of \( a(i,k) + r(i,k) \). If this value is maximized where \( i = k \) then \( i \) itself is an exemplar. Eqs. (2) and (3) show that there is a circular dependency between responsibility and availability. \( a(i,k) \) is initialized to a zero value so that \( r(i,k) \) can be calculated in the first iteration. After this the availabilities are calculated and stored to be ready for the next iteration.

\[
\begin{align*}
 r(i,k) & \leftarrow s(i,k) - \max_{k'=k} \{a(i,k') + s(i,k')\} \tag{2} \\
 a(i,k) & \leftarrow \min \left\{ 0, r(k,k) + \sum_{i' \neq k} \max\{0, r(i',k)\} \right\} \tag{3}
\end{align*}
\]

Responsibility in Eq. (2) can be thought of as a competitive update where the similarity measure between \( i \) and \( k \), \( s(i,k) \), is subtracted by \( k \). That is the greatest similarity value between \( i \) and every other potential exemplar, \( s(i,k') \), plus the corresponding availability value, \( a(i,k') \). It can be seen in Eq. (3) that an availability value will not be greater than zero so this factor will either have no effect on responsibility or it will increase it. Intuitively, if some other candidate exemplar for point \( i \) is less available, the current candidate exemplar being analyzed then becomes more responsible and a better fit.

Availability in Eq. (3) is either zero or less depending on the self responsibility, \( r(k,k) \), of the candidate exemplar and the sum of the positive responsibilities that the candidate exemplar receives from other points. Self responsibility is a measure of how much evidence there is for the point itself to be an exemplar. This measure can be adjusted by a “preference” measure given by the user. By default, all data points are equally suitable as exemplars, so the preferences are set to a common value (e.g., the median of the input similarities in the similarity matrix). It can be seen in Eq. (3) that if \( r(k,k) \) is negative then this will negatively affect the availability of this point, which means this point is less available to be assigned to another cluster.

3. Preliminaries

The list of symbols to be used throughout the rest of this paper is first summarized in Table 1.

**Definition 1.** A mobile cloud (MC) is a graph which is represented by a triplet:

\[
MC = (N, E, T)
\]

where \( N \) refers to the set of nodes, \( E \) refers to a set of edges representing the network bandwidths for data transfer at \( T \) which means the time. In the above definition, due to the instability and heterogeneity of the MC environment, the bandwidth of any two nodes in \( MC \) may be different and variant with the change of the time. In addition, the data transmission distance in the mobile wireless network is limited. So, for any node in the \( MC \), there exits a maximal transmission distance which is called cover radius.

As mentioned above, in order to efficiently facilitate the data transmission and receiving, an access point (AP) needs to be deployed in each node of the wireless network.

**Definition 2.** A node in \( MC \) can be modeled by a triplet:

\[
N_i = (i, pos, R_i)
\]

where \( i \) is the ID number of the node, \( pos \) is the 3-D coordinate values of the node and \( pos = (x,y,z) \), and \( R_i \) is the cover radius.

**Definition 3 (Cover Sphere).** Given a node \( N_i \), its corresponding cover sphere (CS) is a 3-D sphere, formally defined as \( CS(N_i) = \Theta(N_i, Pos, N_i, R_i) \), where \( N_i, Pos \) is the coordinate value of the node, and \( N_i, R_i \) is the cover radius.
Definition 4. The nodes in MC can be logically divided into three categories: the query node \((N_q)\), master nodes \((N_m)\), and slave nodes \((N_s)\), formally denoted as \(N = N_q \cup N_m \cup N_s\), where \(N_m\) and \(N_s\) are composed of \(a\) master nodes \((N_i^m)\) and \(a\) slave nodes \((N_j^s)\), respectively, \(N_i^m\) is the \(i\)th master node, \(N_j^s\) denotes the \(j\)th slave node, for \(i, j \in [1, a]\). As shown Fig. 2, for the master nodes in the MC, they are composed of two kinds of nodes, namely, global master node \((N_{GM})\) and local master node \((N_{LM})\).

Definition 5 (Global Master Node). A global master node, denoted as \(N_{GM}\), is a type of master node which is responsible for query routing. \(N_{GM}\) is composed of \(x\) master nodes \((N_i^m)\), where \(N_i^m\) is the \(i\)th master node, \(N_i^m\) denotes the \(j\)th slave node, for \(i, j \in [1, x]\).

Definition 6 (Local Master Node). A local master node, denoted as \(N_{LM}\), is another type of master node which is responsible for data storage, indexing and data filtering. \(N_{LM}\) is composed of \(x'\) master nodes \((N_i'^m)\), where \(N_i'^m\) is the \(i\)th master node, \(N_i'^m\) denotes the \(j\)th slave node, for \(i, j \in [1, x']\). As defined above, in a MC, after a user submits a query from the query node layer, the global master nodes (totally \(a\) numbers) routes the user query to the corresponding optimal \(N_{LM}\) for further processing. And the (totally \(a\) numbers) \(N_{LM}\)s are responsible for distributed storage of the medical images \((\Omega)\) from different departments and their corresponding indexes. Meanwhile, index-support image filtering in the MIRC process is conducted at the corresponding \(N_{LM}\); the slave nodes can receive the candidate images \((\Omega')\) obtained by the above filtering process. Then, refinement processes (distance computation) of the candidate images in the corresponding slave nodes are conducted. Finally the answer set \((\Omega'')\) is sent back to the query node.

The goal of our proposed method is to query the results in the mobile cloud environment with minimal data communication and CPU cost.

### Table 1

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Omega)</td>
<td>A set of medical images in a hospital</td>
</tr>
<tr>
<td>(\Omega_j)</td>
<td>The medical image set from the (j)th department</td>
</tr>
<tr>
<td>(I_i)</td>
<td>The (i)th medical image and (i \in \Omega)</td>
</tr>
<tr>
<td>(d(\cdot, \cdot))</td>
<td>The distance between two medical images</td>
</tr>
<tr>
<td>(\Omega')</td>
<td>The candidate image set</td>
</tr>
<tr>
<td>(x)</td>
<td>The number of the master nodes or slave nodes</td>
</tr>
<tr>
<td>(x')</td>
<td>The number of the global master nodes</td>
</tr>
<tr>
<td>(x'')</td>
<td>The number of the local master nodes</td>
</tr>
<tr>
<td>(</td>
<td>\cdot</td>
</tr>
</tbody>
</table>

As shown Fig. 2, the three layer architecture of a mobile cloud environment.
4. Enabling techniques

To facilitate efficient similarity query processing in the mobile cloud environment, in this section, we introduce three enabling techniques: (i) a dynamic learning-based data placement scheme; (ii) content-aware and bandwidth-conscious multi-resolution-based data replica selection mechanism; (iii) priority-based image block data robust transmission mechanism; and (iv) a dynamic correlated data caching scheme. Their purpose is to minimize the total communication cost and maximize the query parallelism.

4.1. Learning-based data placement scheme

For existing parallel database systems, the study of data placement is critically important to the efficiency of parallel query processing. Different from traditional parallel systems, the MC is a mobile, wireless, and heterogeneous environment in which users do not stay at a fixed place. Moreover, in a hospital, some departments are usually located at different floors in a building, while others may be in another building. This may raise several challenges to data placement in such a dynamic environment, as explained below.

4.1.1. Motivations

The motivations of the learning-based data placement scheme in master nodes are based on the following key observations:

- Fast data access is very important for the parallel query processing in the Mc environment. So the location of the data server that the users can access is critically important. That is, to minimize the total communication cost, the total distance between all users and their server(s) should be minimized.
- For users in the same department, not all of them are always in the same region (e.g., their department) all the time. For example, some users may appear in some other regions of the hospital due to some other duties. The position coordinate of the centroid, however, will not be dominated by a few users (out of their department), but the most. It motivates us to devise a two-stage approach to obtain an optimal centroid position, as illustrated in Fig. 3.

4.1.2. Mining user moving trajectories

As mentioned above, the efficiency of fast data access is very important to the location-based query in the Mc. So the basic idea of this approach is to find an optimal data placement in the Mc through first mining user movement trajectories.

Definition 7. A user is modeled by a four-tuple:

\[ d_i = (i, Dept, Pos, Tim) \]  

where

- \( i \) denotes the user’s ID;
- \( Dept \) refers to the department that user \( d_i \) is affiliated with;
- \( Pos \) is the coordinate value of \( d_i \), denoted as \( Pos = (x, y, z) \), where \( x, y \) and \( z \) refer to the three coordinate values, respectively;
- \( Tim \) refers to the time.

Fig. 3. Learning-based moving trajectory mining.
Theorem 1. Given $m$ objects: $O_i = (x_i, y_i, z_i)$, where $i \in [1, m]$, there exists a centroid object $C = (x_o, y_o, z_o)$, the total distance $\sum_{i=1}^{m} (x_i - x_o)^2 + \sum_{i=1}^{m} (y_i - y_o)^2 + \sum_{i=1}^{m} (z_i - z_o)^2$ is minimal, iif $x_o = \frac{1}{m} \sum_{i=1}^{m} x_i$, $y_o = \frac{1}{m} \sum_{i=1}^{m} y_i$, and $z_o = \frac{1}{m} \sum_{i=1}^{m} z_i$.

Proof. $\sum_{i=1}^{m} (x_i - x_o)^2 + \sum_{i=1}^{m} (y_i - y_o)^2 + \sum_{i=1}^{m} (z_i - z_o)^2 = mx_o^2 - 2x_o \sum_{i=1}^{m} x_i + \sum_{i=1}^{m} x_i^2 + my_o^2 - 2y_o \sum_{i=1}^{m} y_i + \sum_{i=1}^{m} y_i^2 + mz_o^2 - 2z_o \sum_{i=1}^{m} z_i + \sum_{i=1}^{m} z_i^2$

Let $\sum_{i=1}^{m} (x_i - x_o)^2 + \sum_{i=1}^{m} (y_i - y_o)^2 + \sum_{i=1}^{m} (z_i - z_o)^2 = 0$, then we have

$$x_o = \frac{1}{m} \sum_{i=1}^{m} x_i, \quad y_o = \frac{1}{m} \sum_{i=1}^{m} y_i, \quad z_o = \frac{1}{m} \sum_{i=1}^{m} z_i.$$ (7)

The 1st Step. In Fig. 3(a), suppose that there are nine users in a department. According to Theorem 1, we first calculate the centroid $(C_1)$ of all the nine users by Eq. (8).

$$\begin{align*}
C_1.x &= \frac{1}{|Dep|} \sum_{i=1}^{Dep} d_i \cdot Pos \cdot x \\
C_1.y &= \frac{1}{|Dep|} \sum_{i=1}^{Dep} d_i \cdot Pos \cdot y \\
C_1.z &= \frac{1}{|Dep|} \sum_{i=1}^{Dep} d_i \cdot Pos \cdot z
\end{align*}$$ (8)

where $|Dep|$ means the total number of users in the department Dep.

Definition 8. A virtual sphere region (VSR) is a sphere composed of a centroid $(C_1)$ and a radius $(r)$ in a 3-D space, formally denoted as $\text{VSR} = (C_1, r)$. Given a centroid $C_1$, a VSR centered at $C_1$ and radius $r$ (Ref. the dash blue circle in Fig. 3(b)) can be obtained, where $r$ is a threshold value. So for example, the VSR can be seen as the department region, the users inside the VSR are the candidate ones. The 2nd Step. For all candidate users in the same department, the coordinate values of the final centroid $(C_2)$ of them, shown by a shadow circle region in Fig. 3(c), where $C_2$ is a final center and $r$ is a radius, can be calculated by Eqs. (9) and (10):

$$\begin{align*}
C_2.x &= \frac{1}{|Can|} \sum_{i=1}^{Can} d_i \cdot Pos \cdot x \\
C_2.y &= \frac{1}{|Can|} \sum_{i=1}^{Can} d_i \cdot Pos \cdot y \\
C_2.z &= \frac{1}{|Can|} \sum_{i=1}^{Can} d_i \cdot Pos \cdot z
\end{align*}$$ (9)

$$r = \arg\max \left\{ \frac{|Can|}{\sum_{i=1}^{Can} (|\text{Dis}(C_2, d_i, \cdot Pos)\text{) |} \right\}$$ (10)

where $|Can|$ means the total number of the candidate users in the department Dep.

Combining these two stages, we obtain our learning-based data placement algorithm in Algorithm 1.

Algorithm 1. Learning-based data placement algorithm

Input: $\Omega$: the image set, all users;
Output: the optimal data placement;
1. for each department in a hospital do
2. for all users $d_i$ in the same department do
3. calculate their initial centroid $(C_1)$ according to Eq. (8);
4. end for
5. end for
6. for each user $d_i$ in the same department do
7. calculate the distance between $C_1$ and $d_i$
8. if the distance is less than a threshold value ($v$) then
9. add $d_i$ as a candidate element // represented by blue points in Fig. 3(b)
10. end if
11. end for
12. for each candidate user $d_i$ in the same department do
13. calculate their final centroid $(C_2)$ according to Eq. (9);
14. end for
15. the data server of the department can be placed at the position of $C_2$.  

4.1.3. Adaptive node merge scheme

In the above subsection, we assume that one department corresponds to a master node and a slave node. However, the number of patients and doctors in different departments are actually various due to the difference of the diseases. For example, cold is a very common disease. The number of patients who catch a cold is much more than that of the Tuberculosis. So the number of patients in the department of internal medicine can be larger than that of department of respiratory. The data size of the patient information (e.g., medical records, medical images, etc.) stored in such two departments can be greatly different accordingly.

To economically deploy the master nodes and maximize reuse of resources, we propose an adaptive node mergence (Adm) scheme. In this scheme, local master nodes whose data to be processed is not very large can be merged together. Three factors are considered in the Adm scheme:

- **The data size of the storage**: the data size of the storage of each $N_{LM}$ is analyzed, and the nodes with small size of data storage are merged.
- **Access frequency**: the access frequency of each $N_{LM}$ is mined, and the nodes with low access frequencies are merged.
- **Spatial location**: spatial information is very important for the node mergence, so we try to merge the spatial nearest neighbor $N_{LM}$s.

**Definition 9 (Minimal Cover Sphere).** Given two local master nodes $N_{LM}^i$ and $N_{LM}^j$, their corresponding minimal cover sphere (Mcs) is a 3-D sphere, formally denoted as $\text{Mcs}(N_{LM}^i, N_{LM}^j)$, which is subject to the following conditions:

- Both $\text{CS}(N_{LM}^i)$ and $\text{CS}(N_{LM}^j)$ are contained by the $\text{Mcs}(N_{LM}^i, N_{LM}^j)$;
- The volume of the $\text{Mcs}(N_{LM}^i, N_{LM}^j)$ is minimal;

In Fig. 4, suppose that there are two cover spheres of the $N_{LM}^i$ and $N_{LM}^j$ which are represented by two shadow circles. Their corresponding $\text{Mcs}$ can be denoted by a dash circle. The coordinate values of the center $(N_{LM}^{X})$ of the $\text{Mcs}$ is derived in Theorem 2 below.

Fig. 5(a) illustrates an example of the spatial distribution of five local master nodes, and Table 2 shows the distributions of the above-mentioned three factors of $N_{LM}$s before node mergence. As an example, in Table 2, for a local master node $N_{LM}^{1}$, its nearest neighbor (NN) node is $N_{LM}^{2}$. Assuming their summer values of the access frequencies and the sizes of data storage are less than the threshold values, we can merge the two nodes into a new one $N_{LM}^{S}$ as shown in Table 3. Similarly, $N_{LM}^{i}$ and $N_{LM}^{j}$ can be also merged together.

**Theorem 2.** Given two local master nodes: $N_{LM}^{i}$ and $N_{LM}^{j}$, and their corresponding cover radius: $R_{i}$ and $R_{j}$ the center ($N_{LM}^{X}$) and radius ($R_{x}$) of their corresponding $\text{Mcs}$ can be calculated by the following equations:

\[
\begin{align*}
N_{LM}^{X} \cdot \text{pos} \cdot x &= N_{LM}^{i} \cdot \text{pos} \cdot x + \frac{R_{i}}{R_{i} + R_{j}} \times \left( N_{LM}^{j} \cdot \text{pos} \cdot x - N_{LM}^{i} \cdot \text{pos} \cdot x \right) \\
N_{LM}^{X} \cdot \text{pos} \cdot y &= N_{LM}^{i} \cdot \text{pos} \cdot y + \frac{R_{i}}{R_{i} + R_{j}} \times \left( N_{LM}^{j} \cdot \text{pos} \cdot y - N_{LM}^{i} \cdot \text{pos} \cdot y \right) \\
N_{LM}^{X} \cdot \text{pos} \cdot z &= N_{LM}^{i} \cdot \text{pos} \cdot z + \frac{R_{i}}{R_{i} + R_{j}} \times \left( N_{LM}^{j} \cdot \text{pos} \cdot z - N_{LM}^{i} \cdot \text{pos} \cdot z \right) \\
R_{x} &= \frac{d(N_{LM}^{i}, N_{LM}^{j}) + R_{i} + R_{j}}{2}
\end{align*}
\]

**Proof.** As the volume of the new 3-D minimal cover sphere region is:

\[
\text{Vol(}\text{Mcs}(N_{LM}^{S}, R_{x})) = \frac{4}{3} \pi R_{x}^{3}
\]
To minimize the sphere volume, \( R_x \) needs to be minimized. As shown in Fig. 5, since \( R_x = d(N^X_{NLM}, N^i_{LM}) + R_i = d(N^X_{NLX}, N^j_{LM}) + R_j \), we have

\[
R_x = \frac{d(N^X_{NLM}, N^i_{LM}) + d(N^X_{NLM}, N^j_{LM}) + R_i + R_j}{2}
\]

As \( d(N^X_{NLM}, N^i_{LM}) + d(N^X_{NLM}, N^j_{LM}) \geq d(N^X_{NLM}, N^i_{LM}) \), so if \( d(N^X_{NLM}, N^i_{LM}) + d(N^X_{NLM}, N^j_{LM}) = d(N^X_{NLM}, N^i_{LM}) \), then the value of \( R_x \) is minimized. That is, the volume value is minimal.

Therefore, the center position and radius of the new local master node \( (N^X_{NLM}) \) can be represented as follows:

\[
\begin{align*}
N^X_{NLM} \cdot \text{pos} \cdot x &= N^i_{LM} \cdot \text{pos} \cdot x + \frac{R_i}{R_i + R_j} \times (N^j_{LM} \cdot \text{pos} \cdot x - N^i_{LM} \cdot \text{pos} \cdot x) \\
N^X_{NLM} \cdot \text{pos} \cdot y &= N^i_{LM} \cdot \text{pos} \cdot y + \frac{R_i}{R_i + R_j} \times (N^j_{LM} \cdot \text{pos} \cdot y - N^i_{LM} \cdot \text{pos} \cdot y) \quad \text{and} \quad R_x = \frac{d(N^i_{LM}, N^j_{LM}) + R_i + R_j}{2}
\end{align*}
\]

The adaptive node merge process is detailed in Algorithm 2.

---

**Table 2**

<table>
<thead>
<tr>
<th>( N_{LM} )</th>
<th>Storage size (GB)</th>
<th>Access freq.</th>
<th>Spatial NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N^1_{LM} )</td>
<td>10</td>
<td>10,000</td>
<td>( N^5_{LM} )</td>
</tr>
<tr>
<td>( N^2_{LM} )</td>
<td>2</td>
<td>500</td>
<td>( N^3_{LM} )</td>
</tr>
<tr>
<td>( N^3_{LM} )</td>
<td>5</td>
<td>2500</td>
<td>( N^2_{LM} )</td>
</tr>
<tr>
<td>( N^4_{LM} )</td>
<td>3</td>
<td>1500</td>
<td>( N^5_{LM} )</td>
</tr>
<tr>
<td>( N^5_{LM} )</td>
<td>4</td>
<td>5000</td>
<td>( N^1_{LM} )</td>
</tr>
</tbody>
</table>

**Table 3**

<table>
<thead>
<tr>
<th>( N_{LM} )</th>
<th>Storage size (GB)</th>
<th>Access freq.</th>
<th>Spatial NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N^1_{LM} )</td>
<td>10</td>
<td>10,000</td>
<td>( N^2_{NLM} )</td>
</tr>
<tr>
<td>( N^1_{NLM} )</td>
<td>7</td>
<td>3000</td>
<td>( N^3_{NLM} )</td>
</tr>
<tr>
<td>( N^2_{NLM} )</td>
<td>7</td>
<td>6500</td>
<td>( N^1_{LM} )</td>
</tr>
</tbody>
</table>
Algorithm 2. The ANM algorithm

\textbf{Input:} $\Omega$: the image set, all users;
\textbf{Output:} the optimal data placement;
\begin{enumerate}
\item for each local master node $N_{LM}$ do
\item find its spatial NN node $N_{LM}$
\item calculate the total sum of the data storage sizes of the two nodes;
\item if the value is less than a threshold value ($\varepsilon_1$) then
\item calculate the total sum of the access frequencies of the two nodes;
\item if the value is less than a threshold value ($\varepsilon_2$) then
\item merge the two nodes to a new one;
\item end if
\item end if
\item end for
\end{enumerate}

4.2. Content-aware and bandwidth-conscious multi-resolution-based data replica selection scheme

As mentioned before, since the pixel resolutions of the digital cameras are high, the ordinary image is usually very high (e.g., $2040 \times 2040$), the data size of the image is large accordingly. It is non-trivial to transfer an image with such a big size to the destination nodes, especially in a wireless network environment.

In this subsection, we propose an Content-aware and Bandwidth-conscious Multi-resolution-based data Replica selection scheme (CBMR) by comprehensively considering the relationship of three factors (i.e., image content, network bandwidth and image pixel resolution). As shown in Fig. 6, the basic idea of the CBMR method is that for a same medical image, the image with different pixel resolutions can be transferred according to the variance of the network bandwidth. Specifically, with high network bandwidth, a high-resolution medical image can be transferred in a reasonable short period of time. On the contrary, in order to get a short response time, a low resolution version of the same image can be sent to the destination node with low network bandwidth.

Different from the conventional image, medical images, however, usually reflect the pathological state of the patient’s diseased organs. Since the clinically useful information in medical images is mostly highly localized in small areas of the images, that is, the ratio of pathology bearing pixels to the rest of the images is small. Reducing the pixel resolution of the medical image will probably lead to doctors’ misdiagnosis. Therefore, we need to maintain the resolution of some important areas in the image unchanged, which we called medically useful area (MUA) (e.g., parts A and B in Fig. 7). It is worth mentioning that the MUAs are not whole organ areas like chest and leg, etc., but the lesion parts in them which need to be shown with high pixel resolutions. Although it is not hard to identify and segment the organ areas in the medical images as the background of the images are relatively simple, it is not trivial to automatically and accurately identify the lesion parts of the organs, particularly for some small size lesion parts due to the technical bottlenecks of computer vision. Different from general images, accurate segmentations of the MUAs in the medical images are critically important not only to the correct diagnoses but also to the data transmission costs. Therefore, the MUAs are first detected manually by sophisticated physicians in the preprocessing step. In our future work, we will develop a method to conduct automatic MUA detection.

For the rest part of the images, they can be stored according to the multiple resolutions respectively. Based on the above analysis, the objective of our method is to get a tradeoff between the quality of medical images and the transferring time under different resolution and available network bandwidth.

Let us first denote the resolution of the $i$th image as $R_i \in [x_{LOW}, x_{UPP}]$, where $x_{LOW}$ and $x_{UPP}$ mean the lower bound and upper bound resolutions of the ith image, respectively. Similarly, the bandwidth of the $j$th edge is defined as $E_j \in [y_{LOW}, y_{UPP}]$, where $y_{LOW}$ and $y_{UPP}$ are the lower bound and upper bound bandwidth of the $j$th edge, respectively. Based on an assumption that the image pixel resolution ($R_i$) is proportional to the network bandwidth ($E_j$), so the corresponding image resolution under the current network bandwidth can be derived as follows:

$$R_i = x_{LOW} + \left\lfloor \frac{i \times (x_{UPP} - x_{LOW})}{\Delta} \right\rfloor$$

where $i \in [1, \Delta]$ and $\Delta$ is the granularity (i.e., number of blocks).

Similarly, as the current network bandwidth ($E_j$) is among $y_{LOW}$ and $y_{UPP}$, so it can be represented below:

$$E_j \in \left[ y_{LOW} + \left\lfloor \frac{(i - 1) \times (y_{UPP} - y_{LOW})}{\Delta} \right\rfloor, y_{LOW} + \left\lfloor \frac{i \times (y_{UPP} - y_{LOW})}{\Delta} \right\rfloor \right]$$

As $i$ is an integer, so $i = \frac{E_j - y_{LOW} \times \Delta}{y_{UPP} - y_{LOW}} + 1$, where $\lfloor \cdot \rfloor$ is the integer part of $\bullet$.

Therefore, under the condition of current bandwidth $E_j$, the corresponding pixel resolution of the $i$th image can be derived as follows:
\[ R_i = x_{LOW} + \frac{(E_j - y_{LOW}) \times \Delta}{y_{UPP} - y_{LOW}} + 1 \times \frac{x_{UPP} - x_{LOW}}{\Delta} \]

In Eq. (15), a whole image is regarded as an object to be processed. The pixel resolution of the whole image can be adjusted according to the variance of the network bandwidth. This method, however, will lead to the pixel resolution of the lesion area of the image being so low that the doctor cannot clearly see the lesion area which may cause misdiagnosis. Therefore, in the preprocessing step, as shown in Fig. 8(a), the lesion parts (also called MUA) in the image are firstly identified by drawing two red dash rectangles manually, namely A and B. Fig. 8(b) shows the result of processing. It can be seen that the resolutions of the MUAs A and B in Fig. 8(a) and (b) are unchanged, and the resolution of the rest part (C) in Fig. 8(b), however, is decreased.

Let the area of image \( I_i \) be \( S \), the area of high-resolution parts be \( S_1 \) and the resolution be \( R_{i1} \), the area of the rest part be \( S_2 \) and the resolution be \( R_{i2} \), where \( S = S_1 + S_2 \), then the comprehensive pixel resolution (\( R_i \)) of image \( I_i \) can be derived as:

\[ R_i = \frac{S_1 \times R_{i1} + S_2 \times R_{i2}}{S} \]

(16)

It is worth mentioning that, to ensure the high pixel resolution of the MUAs (A and B), namely, \( R_{i1} \) is fixed in Eq. (16). Replace Eq. (15) with \( R_i \) in Eq. (16), we have

\[ \frac{S_1 \times R_{i1} + S_2 \times R_{i2}}{S} = x_{LOW} + \frac{(E_j - y_{LOW}) \times \Delta}{y_{UPP} - y_{LOW}} + 1 \times \frac{x_{UPP} - x_{LOW}}{\Delta} \]

(17)

then \( R_{i2} \) can be derived as below:

\[ R_{i2} = \frac{S \times [x_{LOW} + \frac{(E_j - y_{LOW}) \times \Delta}{y_{UPP} - y_{LOW}} + 1 \times \frac{x_{UPP} - x_{LOW}}{\Delta}]}{S_2} - S_1 \times R_{i1} \]

(18)
For example, assume that the minimal and maximal pixel resolution of an image is $x_{\text{LOW}} = [200,200]$, $x_{\text{UPP}} = [2040,2040]$. And the bandwidth of the wireless network ranges from 10 MB/S to 100 MB/S, namely, $y_{\text{LOW}} = 10 \text{ MB/S}$, $y_{\text{UPP}} = 100 \text{ MB/S}$.

Additionally, $S = 20 \text{ cm}^2$, $S_1 = 6 \text{ cm}^2$ and the pixel resolution is $R_1 = [2040,2040]$, $S_2 = 14 \text{ cm}^2$.

According to Eq. (18), suppose that the current bandwidth ($E_j$) is 50 MB/S, then the optimal pixel resolution for image data transmission can be derived below:

$$R_2 = \frac{20 \times \left[200,200\right] + \frac{50 - 10}{100 - 10} + 1 \times \frac{2040,2040 - 200,200}{\Delta}}{14} - 6 \times \left[2040,2040\right]$$

(19)

- If $\Delta = 10$, then $R_2 = [725,725]$;
- If $\Delta = 20$, then $R_2 = [594,594]$;
- 

Similarly, if the current bandwidth ($E_j$) is 80 MB/S, then

- If $\Delta = 10$, then $R_2 = [1514,1514]$;
- If $\Delta = 20$, then $R_2 = [594,594]$;
- 

Based on the above analysis, if $\Delta$ is fixed, with the increase of network bandwidth, the optimal transmission pixel resolution is increasing accordingly. Meanwhile, if the bandwidth is fixed, with the increase of $\Delta$, the optimal transmission pixel resolution will decrease accordingly.

4.3. The priority-based image data transmission scheme

Generally speaking, in the state-of-the-art image data transmission schemes, a whole image is transferred as an object. For the medical images with high pixel resolutions, this transmission method, however, will lead to the increase of the failure in the transmission process. Once the transmission failure is occurred, the image needs to be re-transferred which results in a higher transmission cost. To overcome this technical bottleneck, we propose a Priority-Based Image Block Data Robust Transmission Scheme called PBT to support the successive transmission of the large image data.

Definition 10. Each block can be modeled by a seven-tuple:

$$\text{Block} ::= < \text{bid, } x_1, x_2, y_1, y_2, \text{pri, res} >$$

(20)

where

- bid refers to the block number;
- $x_1$ and $x_2$ refer to the x-axis values of the upper-left and lower-right of the block, respectively;
- $y_1$ and $y_2$ refer to the y-axis values of the upper-left and lower-right of the block, respectively;
- pri refers to the transmission priority of the block and $\text{pri} \in [0,1]$;
- res refers to the corresponding pixel resolution of the block;

In this method, for example, an image in Fig. 9 is first partitioned into 20 blocks based on a coarse granularity. After that, the high-resolution region (i.e., part B) of the image is then partitioned into 42 blocks based on a refined granularity. Finally, each block is granted with a different transmission priority.
Specifically, the image in Fig. 9 can be composed of three parts: A, B and C, where A is the rest part of the image except B and C, B is the important region set by user (represented by a red dash rectangle), C is a surrounding region of B (represented by a shadow part). The transmission priority of each block in each part is different, which is shown in Eq. (21):

\[
\text{Block}\cdot\text{pri} = \begin{cases} 
1, & \text{Block belongs to B} \\
0.5, & \text{Block belongs to C} \\
0, & \text{Block belongs to A}
\end{cases}
\] (21)

According the different priorities of the blocks, the image blocks are transferred in terms of the priority in a descending order which not only ensure the robustness of data transmission but guarantee that the important information can be transferred with priority.

4.4. Dynamic correlated data caching scheme

Data cache (or prefetch) is very common in distributed database systems. The aim of the method is to pre-fetch some correlated images with other departments and support fast data access in a distributed environment. However, the side-effect of data redundancy may affect query performance a lot.

Suppose two users \(d_1\) and \(d_2\) belong to different departments A and B, and the distance between A and B is very large. Assume user \(d_1\) in A wants to access some medical images which user \(d_2\) has in B. It is clear that the image access from A to B involves high communication cost. To facilitate faster image access, a part of the medical images in department B which are frequently accessed by users in A can be replicately stored in department A. In this subsection, we propose a learning-based dynamic correlated data caching (DCDC) scheme via analyzing the frequency of each image to be accessed by other departments.

**Definition 11.** A medical image denoted as \(I_i\) can be modeled as a five-tuple:

\[I_i := < i, s\text{Dept}, Acce, PatID, Fea >\] (22)

where
- \(i\) is the ID number of image \(I_i\);
- \(s\text{Dept}\) refers to the source department owning \(I_i\);
- \(\text{Acce} := < d\text{Dept}, Freq >\), where
  - \(d\text{Dept}\) refers to a destination department which accesses image \(I_i\) from \(s\text{Dept}\);
  - \(Freq\) is the times of accesses that users in \(d\text{Dept}\) want to access \(I_i\) in the \(s\text{Dept}\);
- \(\text{PatID}\) is the ID number of the patient \(I_i\) belongs to;
- \(\text{Fea}\) means the feature extracted from the image \(I_i\);

It is worth mentioning that for each image \(I_i\), to obtain its corresponding access frequency, there is an auxiliary data structure for each image shown in Table 4. If the image in \(s\text{Dept}\) is accessed from \(d\text{Dept}\), then add one to its corresponding frequency. Table 4 shows an image access frequency table which is to record the access frequency of individual images.

In Algorithm 3, when the access frequency (freq) that image \(I_i\) in department \(s\text{Dept}\) is accessed by \(d\text{Dept}\) is larger than a threshold value (\(\delta\)), the image \(I_i\) is replicated in \(d\text{Dept}\).

![Fig. 9. The transmission priorities of each image block.](image-url)
Algorithm 3. Learning-based DCDC algorithm

\textbf{Input}: $\Omega$: the image set, all users;  
\textbf{Output}: the optimal data redundancy;

1. \textbf{for} each department in a hospital \textbf{do}
2. \textbf{for} each image $I_i$ in the same department \textbf{do}
3. \textbf{if} $I_i/\text{Access freq} > \delta$ \textbf{then}
4. \hspace{1em} send the image $I_i$ to department $d_{\text{Dept}}$ as a copy image;
5. \textbf{end if}
6. \textbf{end for}
7. \textbf{end for}

5. Two index schemes

In order to reduce major processing (i.e., I/O, CPU and commutation) cost and speed up the query efficiency, we present in this section two indexing schemes which are (1) \textit{query routing index in the global master nodes}, and (2) \textit{major and minor index schemes at the local master node level}.

5.1. Query routing index

As mentioned before, a doctor may sometimes need to obtain some medical images from other departments ($d_{\text{Dept}}$) in his department ($s_{\text{Dept}}$). When he submits a query request to the master node level, a query routing process is first performed in the $N_{GM}$ to locate the corresponding $N_{LM}$ (i.e., $s_{\text{Dept}}$) in which the candidate images may be in. If the candidate ones are not in the $N_{LM}$, the query will be redirected to other $N_{LM}$ of the corresponding department (i.e., $d_{\text{Dept}}$).

5.1.1. Optimal number of $N_{GM}$ s

As introduced in Section 3, the AP-cluster [18] is an effective and optimal clustering method in which optimal cluster centers can be obtained based on the analysis of the distance matrix data. So to obtain an optimal number of $N_{GM}$s, we adopt the AP-cluster.

For any two $N_{LM}$s (e.g., $N_{LM}^i$ and $N_{LM}^j$), the distance between them can be calculated in Eq. (23).
Based on Eq. (23), we can have a distance matrix as an input of the clustering algorithm. Fig. 10 shows an example of the clustering result of the eight local master nodes. The clusters can be represented by dash circles in this figure.

5.1.2. Tree-based organization scheme for the master nodes

After the \( N_{LM} \)s are grouped to some clusters, the center of each cluster corresponds to a \( N_{GM} \). To facilitate an efficient query, how to organize such master nodes is a very important problem.

As shown in Fig. 11, we propose a tree-based organization method for the master nodes. In this method, the \( N_{LM} \)s are organized as a tree structure in which \( N_{LM} \)s are at the leaf node level. It is worth mentioning that the root node \( (R_n) \) in this tree structure is also a \( N_{GM} \). According to the difference of the departments the query involved, there are two types.

(i) If a query is submitted from the sDept (e.g., \( N_{LM} \)) which only involves images from the sDept, then the routing sequence is: \( R_N \rightarrow N_{LM} \rightarrow N_{LM} \);
(ii) If a query is submitted from the sDept (e.g., \( N_{LM} \)) which involves some images not only from \( N_{LM} \) but from dDept (e.g., \( N_{LM} \)), then the routing sequence is: \( R_N \rightarrow N_{GM} \rightarrow N_{LM} \rightarrow N_{LM} \).

5.1.3. The Indexing Scheme

In this subsection, we propose a query routing index (QRI). The aim of the query routing index is to quickly find the locations of the node(s) that the query may involve.

Based on the above analysis, the query routing index key at the root node level can be represented as:

\[
qKey(I) = c_0 \times sID + dID \tag{24}
\]

Similarly, the query routing index key at the second node level can be derived as below:

\[
qKey(II) = c_0 \times sID + dID + PatID/c_1 \tag{25}
\]

where \( c_0 \) and \( c_1 \) are two constants, sID and dID refer to the ID numbers of the source department and the destination one, respectively.

It is worth mentioning that the above two keys can be indexed by an advanced B*-Tree. As shown in Fig. 12, for the QRI(I), each node consists of three elements: (1) IP address to the corresponding \( N_{GM} \), (2) key value (see Eq. (24)), and (3) pointer to the...
next node. For the QRI(II), each node also consists of three elements: (1) IP address to the corresponding NLM, (2) key value (see Eq. (25)), and (3) pointer to the next node.

The query routing process is performed as follows: when a user d1 in sDept submits a query which involves retrieving some images from dDept. The query is first sent to the RN, a QRI(1)-support query location can be conducted to route the query to the corresponding NLM, in which the key range is \([c_0 \times \text{slID} + \text{dID} - e, c_0 \times \text{slID} + \text{dID} + e]\), where e is a small constant and \(e = 1/10\). Then, at the second node level, on the support of the QRI(II), the query is sent to the R, query location can be performed to the corresponding NLM, in which the key range is \([c_0 \times \text{slID} + \text{dID} + \text{PatID}+ c_1 - e, c_0 \times \text{slID} + \text{dID} + \text{PatID}+ c_1 + e]\). If no answer returned, then the query is re-directed to the dDept for further retrieval, else the answer images from dDept cached in sDept can be directly sent back to the user. Algorithm 4 detailed the whole process.

\[
\text{Algorithm 4. Qroute}(I_q,r)
\]

\[
\begin{align*}
\text{Input: } I_q; & \quad \text{the query image.} \\
\text{slID, dID: } & \quad \text{the ID numbers of the source and destination departments, respectively,} \\
\text{PatID: } & \quad \text{the ID number of the patient;} \\
\text{Output: } & \quad \text{the answer images} \\
1. & \quad S_1 = S_2 = \emptyset; \\
2. & \quad \text{A user } d_1 \text{ in sDept submits a query;} \\
3. & \quad \text{The query is first sent to the } R_N; \\
4. & \quad S_1 \leftarrow BRange[c_0 \times \text{slID} + \text{dID} - e, c_0 \times \text{slID} + \text{dID} + e]; \\
5. & \quad \text{The query is routed to the corresponding } N_{LM}; \\
6. & \quad S_2 \leftarrow BRange[c_0 \times \text{slID} + \text{dID} + \text{PatID}+ c_1 - e, c_0 \times \text{slID} + \text{dID} + \text{PatID}+ c_1 + e]; \\
7. & \quad \text{if } (S_2==\text{NULL}) \text{ then} \\
8. & \quad \text{the query is re-directed to the } dDept \text{ for further retrieval;} \\
9. & \quad \text{else} \\
10. & \quad \text{the answer images from } dDept \text{ cached in } sDept \text{ can be directly sent back to the user;} \\
11. & \quad \text{end if}
\end{align*}
\]

5.2. Major and minor index schemes

As defined in Section 3.1, at the master node level, the image data and its index are stored at the local master nodes. According to the data source from different departments, the index can be divided into two categories: major index and minor index.

5.2.1. The major index

As mentioned above, suppose that for the images stored in department sDept, they can be divided into two categories. One is from the same department, another is from other ones. So in this subsection, we will first focus on the index scheme for the image data in the department sDept, which is called the major index.

**Definition 12** (Major Index). The major index (MAI) for the ith department is to index the high-dimensional image data from the same department. For the major index, we adopt the iDistance [21] to index the high-dimensional features extracted from the image data.

5.2.2. The minor index

**Definition 13** (Minor Index). The minor index (MI) for the slIDth department is to index the high-dimensional image data from the dIDth department, where \(slID = dID\). For the image from m other departments (i.e., dDept), it can be denoted as \(I_{ij}\), where \(i = slID\) and \(j = dID \in \{1, m\}\). They are first grouped into \(K\) clusters by the well-known k-Means clustering algorithm, then the MII index key of the \(I_{ij}\) can be derived as:

\[
ikey(I_{ij}) = c_0 \times dID + c_1 \times \text{PatID} + \text{Cid} + \text{dis}(I_{ij}, O_k)/c_2
\]

(26)

where \(c_0, c_1\) and \(c_2\) are three constants, the definitions of \(dID\) and \(\text{PatID}\) are the same to that of in Eq. (25).

The query process is conducted as follows: when a user \(d_1\) in sDept submits a query which involves retrieving some images from the dIDth department with patient ID. The key range is \([\text{left}, \text{right}]\), where \(\text{left} = c_0 \times dID + c_1 \times \text{PatID} + \text{Cid} + \text{dis}(I_{ij}, O_k)/c_2\), and \(\text{right} = c_0 \times dID + c_1 \times \text{PatID} + \text{Cid} + R_k\). Algorithm 5 detailed the whole process.
Algorithm 5. MinorQ \((I_q,r,dID,PatID)\)

**Input:** \(I_q\): the query image,
- \(dID\): the ID number of the destination department,
- \(PatID\): the ID number of the patient;
**Output:** the answer images from the \(dID\)th department

1. \(S \leftarrow \phi\);
2. for each cluster sphere \(\Theta(O_j,R_j)\) and \(j \in [1,T]\)
3. if \(\Theta(O_j,R_j)\) contains \(\Theta(I_q,r)\) then
4. \(S \leftarrow S \cup \text{Search}(I_q,r,j,dID,PatID)\);
5. end loop
6. else if \(\Theta(O_j,R_j)\) intersects \(\Theta(I_q,r)\) then
7. \(S \leftarrow S \cup \text{Search}(I_q,r,j,dID,PatID)\);
8. end if
9. end for
10. for each image \(I_{ij} \in S\) do
11. if \(\text{Sim}(I_q,I_{ij}) > r\) then \(S \leftarrow S \setminus I_{ij}\); // the refinement stage
12. end for
13. return \(S\); // return answer images

Search \((I_q,r,k,dID,PatID)\)

14. \(S_1 \leftarrow \phi\);
15. left \(- c_0 \times dID + c_1 \times PatID + k \times (\text{dis}(I_q,O_k) - r)/c_2\)
right \(- c_0 \times dID + c_1 \times PatID + k \times R_k/c_2\);
16. \(S_1 \leftarrow \text{BRSearch}[\text{left},\text{right}];\) // the filtering step
17. return \(S_1\); // return the candidate character set

6. The MIRC algorithm

With the support of the above enabling techniques, a MIRC query can be efficiently conducted in the mobile cloud environment. The whole query process can be composed of three stages:

6.1. Query routing

In the first stage, a user submits a query request (namely query image \(I_q\) with a department information and radius \(r\)) to the master node level \(N_m\), as described in Algorithm 4, if the query involves retrieving images from other departments, then the query routing process is conducted to locate the query to the corresponding \(NLM\) (i.e., \(sDept\) or \(dDept\)).

6.2. Medical image data filtering process

The second stage is called medical image data filtering process. As mentioned in the first stage, once the query request is disseminated to the corresponding local master node \(N_{LM}\), there are three cases according to the data source:

- For the first case, only images from the \(sDept\) are involved in the query. The irrelevant images in the corresponding \(N_{LM}\) are filtered quickly by using the major index of the \(N^{SD}\).
- For the second case, only images from the \(dDept\) are involved in the query. In the \(sDept\), the cached images from the \(dDept\) are first retrieved by the minor index of the \(sDept\). Then, other images from the \(dDept\) can be obtained quickly by using the major index of the \(N^{ID}\).
- For the third case, images from both of the \(sDept\) and \(dDept\) are involved in the query. First, the irrelevant images in the \(N^{SD}\) are filtered quickly by its major index; then check if some similar images are contained in the cached data via using the minor index in the same node. Finally, other relevant images in the \(N^{ID}\) are obtained quickly by the major index of the \(dID\)th \(N_{LM}\).

In the local master node \(N_{LM}\), an input buffer called \(IB\) is created for caching the images in \(\Omega(i)\), where \(\Omega(i)\) refers to the images in the \(i\)th master node and \(i\) can be \(sID\) or \(dID\). Meanwhile, an output buffer \(OB\), which is used to store the candidate image set \(\Omega'(i)\), is also created. Once the data size of the candidate images in \(OB\) reaches the size of a package, the candidate images are transferred to the slave node with the same department through the package-based data transfer mode.
are computed in the slave node. If and a radius is set, which is used to store the answer images temporarily. If the data size of the answer image set in the package-based manner.

Specifically, in the transfer manner.

In Algorithm 6, the routine \( \text{iDistance}(l_q, r, i) \) returns the candidate image set in the \( j \)th cluster (cf. [21]).

6.3. Refinement

In the final stage, the distances between the candidate images and the query image \( l_q \) are computed in the slave node. If the distance is less than or equal to \( r \), then the candidate image is sent to the query node \( N_q \) in the package-based manner. Specifically, in the \( j \)th slave node \( N_j \), we need to set an input buffer \( IB \) and the memory \( M \) for the candidate image set \( \Omega \). Additionally, an output buffer \( OB \) is set, which is used to store the answer images temporarily. If the data size of the answer image set in \( OB \) equals to the package size, then the answer images are sent to the query node \( N_q \) in the package-based transfer manner.

Algorithm 6. \( \text{MiF}(l_q, r) \)

\hspace*{1cm} **Input:** \( \Omega(sID) \): the sub image set in the \( sID \)th \( N_{LM} \)
\hspace*{1cm} \( \Omega(dID) \): the sub image set in the \( dID \)th \( N_{LM} \)
\hspace*{1cm} \( \Phi(l_q, r) \): the query sphere.
\hspace*{1cm} **Output:** the candidate image set \( \Omega'(sID) \) or \( \Omega'(dID) \)

1. \( \Omega'(sID) = \Omega'(dID) = \Phi \); /* initialization*/
2. Case 1:
3. \hspace*{1cm} for \( i=1 \) to \( T \) do
4. \hspace*{1cm} \hspace*{1cm} \( \Omega'(sID) = \Omega'(sID) \cup \text{iDistance}(l_q, r, i) \);
5. \hspace*{1cm} \hspace*{1cm} \( \Omega'(sID) \) is cached in the output buffer \( OB_{sID} \);
6. end for
7. Case 2:
8. \hspace*{1cm} for i=1 to T do
9. \hspace*{1cm} \hspace*{1cm} \( \Omega'(dID) = \Omega'(dID) \cup \text{iDistance}(l_q, r, i) \);
10. \hspace*{1cm} \hspace*{1cm} \( \Omega'(dID) \) is cached in the output buffer \( OB_{dID} \);
11. end for
12. Case 3:
13. \hspace*{1cm} for i=1 to T do
14. \hspace*{1cm} \hspace*{1cm} \( \Omega'(sID) = \Omega'(sID) \cup \text{MinorQ}(l_q, r, dID, PatID) \);
15. \hspace*{1cm} \hspace*{1cm} \( \Omega'(sID) \) is cached in the output buffer \( OB_{sID} \);
16. end for
17. \hspace*{1cm} for i=1 to T do
18. \hspace*{1cm} \hspace*{1cm} \( \Omega'(dID) = \Omega'(dID) \cup \text{iDistance}(l_q, r, i) \);
19. \hspace*{1cm} \hspace*{1cm} \( \Omega'(dID) \) is cached in the output buffer \( OB_{dID} \);
20. end for
21. return \( \Omega'(sID) \) or \( \Omega'(dID) \)

In Algorithm 6, the routine \( \text{iDistance}(l_q, r, i) \) returns the candidate image set in the \( j \)th cluster (cf. [21]).

Algorithm 7. \( \text{MiR}(\Omega', l_q, r) \)

\hspace*{1cm} **Input:** the candidate image set \( \Omega'(sID) \) or \( \Omega'(dID) \), a query image \( l_q \) and a radius \( r \)
\hspace*{1cm} **Output:** the answer image set \( \Omega''(sID) \) or \( \Omega''(dID) \)

1. \( \Omega''(sID) = \Omega''(dID) = \Phi \);
2. \hspace*{1cm} for \( l_i \in \Omega'(sID) \) do
3. \hspace*{1cm} \hspace*{1cm} if \( d(l_i, l_q) \leq r \) then \( \Omega''(sID) = \Omega''(sID) \cup l_i \);
4. \hspace*{1cm} end for
5. \hspace*{1cm} \( \Omega''(sID) \) is cached in the output buffer \( OB_{sID} \);
6. \hspace*{1cm} for \( l_i \in \Omega'(dID) \) do
7. \hspace*{1cm} \hspace*{1cm} if \( d(l_i, l_q) \leq r \) then \( \Omega''(dID) = \Omega''(dID) \cup l_i \);
8. \hspace*{1cm} end for
9. \hspace*{1cm} \( \Omega''(dID) \) is cached in the output buffer \( OB_{dID} \);

Algorithm 8 shows the detailed steps of our proposed \text{MiRC} algorithm. Note that, the candidate images in each master node are stored in the buffer temporarily before they are sent to the slave nodes. When the number of images in the buffer reaches the size of a “package”, they are sent to the corresponding slave nodes. Next, the answer image set is sent to the buffer of the slave node; finally they are sent to the query node \( N_q \).
Algorithm 8. MIRCSearch \((I_q,r)\)

**Input:** a query image \(I_q\), \(r\)

**Output:** the query result \(S\)

1. \(r \leftarrow 0, \Omega' \leftarrow \Phi, \Omega'^{r} \leftarrow \Phi; /\text{initialization}/\)
2. a query request is submitted to the master node \(N_m\) with department and patient information;
3. if the query involves images from other departments then
   4. a query routing process is first performed; //see Algorithm 3
   5. end if
6. \(\Omega' \leftarrow \text{MIF}(I_q,r)\);
7. the candidate images in \(\Omega'\) are sent to the corresponding slave node;
8. \(\Omega'^{r} \leftarrow \text{MIR}(\Omega',I_q,r)\);
9. the answer image set \(\Omega'^{r}\) is sent to the query node \(N_q\);
10. return \(\Omega'^{r}\)

7. Experimental results

To verify the efficiency of the proposed MIRC method, we conduct simulation experiments to demonstrate the query performance.

The retrieval system is run on Android platform [36] and implemented by Java language. The backend system is simulated by the Amazon EC2 [37]. Each unit is a small instance of EC2 with 2.7 GHz Xeon processor, 2.0 GB memory and 1 TB hard disk. The units are connected via 1 Gbps network links. The number of nodes in our system varies from 10 to 100. The maximum data communication rate is 150 Mbps in the wireless network.

The medical image datasets we used are from two ways: (1) 100 k medical images are downloaded from Medical image archive (http://www.ece.ncsu.edu/imaging/Archives/ImageDataBase/Medical/index.html), (2) 800 k medical images are obtained from the Hangzhou First People’s Hospital from 1990 to 2010. So the total number of the images in this data set is 900 k, in which 64-D color histogram features are extracted, and the value range of each dimension is between 0 and 1.

To test the effectiveness of our proposed method in different kinds of medical images, the medical image dataset is first divided into three categories in terms of the organs such as: (a) lung image; (b) leg image; and (c) heart images; Meanwhile, the image dataset can also be divided into three categories in terms of data formats such as: (a) X ray image; (b) CT image; and (c) MRI image. Each category consists of two image groups: (1) the pixel resolutions of all images in the first group are original; and (2) the pixel resolutions of the images in the second group are mixed. Additionally, to evaluate the effect of data size on the image retrieval and transmission performances, we have synthesized five groups of the medical image data in which the data size of each image are 1 MB, 5 MB, 10 MB, 50 MB and 100 MB, respectively.

For the index part of the system, iDistance [21] is adopted to support quick filtering of medical images deployed at the master node level.

7.1. A prototype retrieval system

We have implemented an online interactive retrieval system for medical images which is illustrated in Fig. 13. In the above part of Fig. 13, when user submits an example image, the candidate images are quickly retrieved by the system with the aid of the index. The right part of the figure is the query result in which the similarity values of the answer image images are given with respect to the query one.

Fig. 14 shows an example of the backend interface of offline medical image processing. The leg image in this figure includes three medically useful areas (MUA) which are identified by three green rectangles. After that, image blocking and pixel resolution adjustment processing are automatically conducted sequentially.

7.2. Comparison of the retrieval accuracy

In the first experiment, we testify the effectiveness of our retrieval method by using the above mentioned two different datasets, respectively. The first dataset includes three categories such as lung images, leg images, and heart images in terms of the different organs; while the second dataset includes X ray images, CT images, and MRI images with respect to different image formats. Each category consists of two image groups: (1) the pixel resolutions of all images in the first group are original; (2) the pixel resolutions of the images in the second group are mixed. Our approach is to perform \(k\)-NN search of the query image with mixed resolution while the conventional one submits the query one with original resolution. Denoting the set of ground-truth as \(rel\), and the set of results returned by a \(k\)-NN search as \(ret\), the recall and precision achieved by this retrieval method are defined as:
recall = \frac{|rel \cap ret|}{|ret|}, \quad precision = \frac{|rel \cap ret|}{|rel|} 

Fig. 13. One retrieval example.

Fig. 14. An backend interface of offline medical image process.

Fig. 15(a)–(c) illustrate recall-precision curves for the performance comparisons between our approach and that of the conventional one. In particular, it compares the average retrieval result (the average precision rate under the average recall rate) of 10 images with the same organ (i.e., lung, leg, and heart) that are randomly chosen from the database. The figures show that the retrieval effectiveness of the conventional approach is similar to that of our proposed method. The change of pixel resolution of the medical image of a same organ is not vulnerable to the retrieval accuracy. This is because different
resolutions of a same image may not affect the differences of the visual feature extraction of the image with different resolutions significantly.

Similarly, in Fig. 16(a)–(c), the average retrieval accuracies of images with the different formats (i.e., X-ray image, CT image, and MRI images) are shown by comparing the recall-precision ratios between our approach and that of the conventional one. For each format, 10 images with the same format are randomly selected from the database as query ones. We found that the retrieval effectiveness of the conventional approach is similar to that of our proposed method. For each format, since in most cases different formats of a same image also may not affect the differences of the visual feature extraction of the image with different formats significantly, so the image format change is not vulnerable to the retrieval accuracy.

7.3. Effect of the CBMR

In this experiment, we study the effect of the CBMR scheme on the performance of the MIRC query processing. Method 1 does not adopt the CBMR and method 2 uses it. Fig. 17(a) shows when $r$ is fixed and the bandwidth is relatively stable, the total response time using the method 2 is superior to that of method 1. Meanwhile, with the condition that the bandwidth is stable and $r$ is increasing gradually, the performance gap becomes larger. This is because the data size of the candidate images to be transferred is increasing so rapidly that the images cannot be sent to the destination nodes quickly.

Moreover, Fig. 17(b) shows that if $A$ is fixed, with the increase of the network bandwidth, the pixel resolution of the images to be transferred can be also increased. The reason is the same to the theoretical analysis in Section 4.2.

Fig. 17(c) demonstrates that if the bandwidth is fixed, when $A$ is increasing, the optimal pixel resolution of image to be transferred is reduced.

7.4. Effect of the PBT

In this experiment, we first test the efficiency of the two transmission schemes: (1) Our proposed PBT method and (2) the progressive image transmission (PIT) [9]. In Fig. 18, when the image data size is from 20 MB to 100 MB, the transmission time of the PBT method is gradually increasing and better than that of the PIT one. This is because compared with the mixed pixel resolution-based image reconstruction of the PBT scheme, the PIT adopts original resolution to reconstruct the images. The data size to be transferred by the PIT is larger than that of the PBT.
To evaluate the effect of data size on the transmission robustness, we then adopted five groups of image data in which the data size of each image are 5 MB, 10 MB, 20 MB, 50 MB and 100 MB. The transmission robustness (TR) can be defined in Eq. (28).

$$TR = \frac{\text{Number of successful data transmissions}}{\text{Total number of data transmissions}} \quad (28)$$

As shown in Fig. 19, with the increase of data size, the successful data transmission ratio (STR) is 100% by using the image blocking technique. For the data transmission without adopting this PBT method, when the transmission data size is less than 10 MB, the successful transmission ratio is 100%, however, if the data size is 20 MB, the average STR is decreased to 87%. And if the data size is larger than 50 MB, the average STR is zero since it is hard to transfer such a large image successfully. Based on the experimental result, to guarantee a high successful data transmission ratio, it is possible to transfer a large image only through the image blocking method in a limited network bandwidth.

7.5. Effect of the DCDC

This experiment studies the effect of the dynamic correlated data caching (DCDC) scheme studied in Section 4.4 on the MIRC query processing. We run the experiment 100 times with 100 query requests randomly generated and obtain an average query response time.

Again, we compare two methods here: method 1 does not adopt the DCDC algorithm and method 2 uses it. As shown in Fig. 20, when the bandwidth is relatively stable, the total response time using the method 2 is superior to that of method 1 with different r. As r is increasing gradually, the performance gap becomes larger since the communication cost of the correlated candidate images to be transferred is also increasing, so rapidly that the images cannot be sent to the destination nodes quickly.
7.6. Effect of the QRI

This experiment tests the effect of the query routing index on query performance. Here, method 1 adopts the QRI index, while method 2 does not adopt it (i.e., it sequentially searches each local master node to identify the corresponding one). Suppose that the data size and the network bandwidth are fixed, when the number of nodes (including master and slave nodes) varies from 20 to 100 (Ref. Fig. 21), the gap between the response time of the two approaches becomes larger and larger. This is because with the support of the QRI, the response time for locating the corresponding node is much faster than that of no index, especially for a large number of master nodes.

7.7. Effect of data filtering

In this experiment, we study the effect of the data filtering process (see Algorithm 6) with the support of major and minor indexes on the query. We assume that the data size and the number of master (slave) nodes are fixed. As shown in Fig. 22, when \( r \) increases from 0.2 to 1, the query response time is also gradually increasing. This is because with the increase of \( r \), the search region in the high-dimensional spaces becomes larger, which leads to the fact that the number of candidate images is increasing accordingly. Still, the approach using data filtering beats the one not using data filtering under all the different \( r \) values.

7.8. Effect of tree-based organization method for master nodes

In the last experiment, we proceed to study the effect of tree-based master node organization method on the MIR query processing. We will compare two search methods: (1) our proposed tree-based scheme; (2) sequential scan-based one in which...
the corresponding $N_{lm}$ is located via searching the $2^m N_{lm}$. To obtain an average query response time, 100 query requests are randomly generated in which each query is run 10 times.
Method 1 adopts the tree-based overlay structure, while method 2 uses a sequential scan to locate the corresponding master node. In Fig. 23, when the number of master nodes is fixed and the bandwidth is relatively stable, the total response time using the method 2 is larger than that of method 1. Meanwhile, with the condition that the bandwidth is stable and the number of master nodes is increasing gradually, the performance gap becomes larger since the searching time for the corresponding master node can be reduced on support of the tree-structure overlay network.

8. Conclusions and future work

In this article, we have presented a system framework of the mobile-cloud-based similar (MC) query processing of the large-scale medical image dataset, which specifically caters for the different bandwidth of nodes in the mobile cloud network and the complexity of the medical images. Four enabling techniques, namely, learning-based adaptive data placement scheme, content-aware and bandwidth-conscious multi-resolution-based image data replica selection scheme, priority-based image block data robust transmission algorithm and dynamic correlated data caching scheme are proposed to reduce the data communication cost. The experimental studies indicate that the proposed MC method is more suitable for the large-scale medical image retrieval in minimizing the network communication cost and maximizing the parallelism in I/O and CPU.

In our future work, we will focus on the following:

1. As the medically interest areas are manually detected by the sophisticated physicians in the preprocessing step, to reduce the burden of the manual detections and segmentations, in our future work, we will develop a method to automatically or semi-automatically detect the medically interest areas in each medical image;
2. Facing the challenges of query intensive applications, to further improve the whole throughput of the queries, we will investigate a pipeline-based distributed query optimization technology to further speed up the query efficiency in the mobile cloud environments;
3. To extend our work to the cross-region domain, we will devise further query optimization methods over different regions which are inter-connected by a wide area network (WAN);
4. Last but not the least, we will combine both semantic and content information of medical images to offer unified query processing in the MC network.

Acknowledgements

This paper is partially supported by the Program of National Natural Science Foundation of China under Grant Nos. 61003074, 61103229, 61272188 and 71072172; the Program of Natural Science Foundation of Zhejiang Province under Grant Nos. LY13F020008, and LY13F020010. The Science & Technology Innovative Team of Zhejiang Province under Grant No. 2012R10041-06.

References


**Further reading**