Pers Ubiquit Comput (2013) 17:1817–1834
DOI 10.1007/s00779-013-0660-4

ORIGINAL ARTICLE

Personalized multi-modality image management and search for mobile devices

Kun Li · Changyun Zhu · Qin Lv · Li Shang · Robert P. Dick

Received: 25 December 2011 / Accepted: 21 September 2012 / Published online: 17 April 2013

Abstract Mobile devices are quickly becoming a primary medium for personal information gathering, management, and sharing. Managing personal image data on mobile platforms is a challenging problem due to large data set size, content and context diversity, heterogeneous individual usage patterns, and resource constraints. This article presents a user-centric system, called iScope, for personal image management and sharing on mobile devices. iScope uses multi-modality clustering of both content and context information for efficient image management and search, and online learning techniques for predicting images of interest. It also supports distributed image search among networked devices while maintaining the same intuitive interface, enabling efficient information sharing among people. We have implemented iScope and conducted infield experiments using networked Nokia N810 portable Internet tablets. Energy efficiency was a primary design focus during the design and implementation of the iScope search algorithms. Experimental results demonstrate that iScope improves search time and search energy by 4.1× and 3.8× on average, relative to browsing.

Keywords Retrieval models · Management · Energy · Performance

1 Introduction

Personal, portable communication, and computation devices are now part of hundreds of millions of lives, often in the form of smartphones. Emerging mobile applications and services are the main driving forces of the prevalence of personal mobile systems. From Daniel Henderson's 1993 prototype, *intellect*, which can receive and display images and video media [1], to the first photograph taken by Philippe Kahn in 1997 using a camera phone, the functionality and adoption of personal portable devices have continuously increased. Today's personal portable devices, such as the iPhone from Apple, Blackberry from RIM, and Android phone from Google, have integrated a variety of system functions, such as global positioning system (GPS), cameras, sensors, large touch screens, and easy-to-use interface. Global mobile phone subscriptions have reached 5.9 billion in 2011 [2]. Users are able to capture information anywhere and anytime. Mobile devices are heavily used for information sharing and social interaction.

Mobile devices are the first-level interface for capturing and sharing multimedia data such as images. They are therefore a natural image data management platform. Managing image data on mobile devices, however, is a challenging problem. A picture may be worth a thousand

K. Li (✉) · L. Shang
ECEE Department, University of Colorado Boulder, Boulder, CO, USA
e-mail: kun.li@colorado.edu

L. Shang
e-mail: li.shang@colorado.edu

C. Zhu
ECE Department, Queen's University, Kingston, Canada
e-mail: changyun.zhu@ece.queensu.ca

Q. Lv
CS Department, University of Colorado Boulder, Boulder, CO, USA
e-mail: qin.lv@colorado.edu

R. P. Dick
EE Department, University of Michigan, Ann Arbor, MI, USA
e-mail: dickrp@eecs.umich.edu

words, but without knowing the words, search can be very difficult. Manual image annotation is tedious and time consuming. *Content-based image retrieval (CBIR)*, which automatically extracts representative features from the raw data and uses the extracted features to locate content of interest, largely automates managing and exploring image data [3]. Albeit recent progress in CBIR, managing image data on personal mobile systems is challenging due to large data set size, content diversity, heterogeneous usage patterns, and resource constraints.

- **Energy-induced constraints:** Energy consumption is a foremost design concern in battery-powered mobile systems. Scarce energy resources largely limit the performance and functionality of software applications running on portable devices. Existing CBIR techniques have high computation complexity and storage requirements. User interaction and communication bring high time and energy cost during personal image search. Energy-induced design constraints introduce serious challenges to the design and implementation of image management systems on personal mobile devices.
- **User-specific search scenarios:** Unlike general-purpose search techniques developed for the World Wide Web, image management on personal mobile devices is a highly personal, user-centric task. Typically, a mobile device has one owner; the data captured and stored on a device depend on its owner's interests. Search patterns are also user specific. It is essential to adapt to users' unique data interests and search patterns to improve search performance and energy efficiency.
- **Distributed data sharing:** Supporting data sharing in distributed mobile environments requires efficient, distributed data management, and search techniques. Communication-induced energy overhead is of great importance in distributed mobile environments. We investigate the time and energy overhead of remote image retrieval and propose collaborative search and metadata caching techniques to allow efficient image sharing and retrieval in distributed mobile environments.

In this work, we describe iScope, a personal content management platform. iScope is a user-centric design targeting energy-constrained distributed mobile environments. It leverages both personal context information and efficient content search techniques, as well as online learning techniques, to deliver personalized, energy-efficient content search services. It provides a collaborative search environment, enabling distributed image search on mobile devices, thus facilitating information discovery and social interaction. We have implemented a prototype of iScope and conducted infield experiments using Nokia N810 portable Internet tablet devices.

A preliminary version of this work was previously published [4]. This article makes further improvements by incorporating incremental hierarchical clustering for mutli-modality clustering with significant speedup and high quality. What is more, this article analyzes and identifies an inherent retrieval pattern with regard to users' memory, which is particularly appealing for daily information retrieval tasks performed by individual users. We discuss the impact of the pattern and how it can benefit users with large amounts of information to manage.

The rest of this article is organized as follows. Section 2 gives an overview of the iScope system architecture. Section 3 presents multi-modality image data management. Section 4 conducts resource characterization of portable platform and investigates the resource usage of image search process. Sections 5 and 6 describe the designs and benefits of personalized and collaborative image search. Sections 7, 8, and 9 describe the experimental results with the iScope prototype, survey related work, and conclude.

2 iScope: overview of system architecture

This section presents an overview of iScope's system architecture. As illustrated in Fig. 1, iScope consists of the following key components.

- *Multi-modality data management:* Personal image data contain a rich set of content information (e.g., color, texture, and shape) and user-specific context information (e.g., location, time, and ownership). In iScope, the context and content information of personal image data are used in unison to enable efficient image management. Images are partitioned based on content features and context metadata. The proposed incremental hierarchical clustering-based multi-modality data management design allows efficient traversal of the data set across different feature dimensions and resolutions, enabling efficient management of personal data sets and run-time user queries (Sect. 3).
- *User-centric adaptive image search:* iScope offers personalized image search by leveraging both content-based search algorithms and user-specific context information. Users differ from each other on image interests and performance expectations. iScope incorporates run-time learning techniques for online prediction and adaptation of the search process based on implicit user feedback, improving search quality and minimizing search costs, e.g., energy consumption (Sect. 5).
- *Distributed collaborative search:* iScope supports remote image search and metadata caching among distributed image data sets spanning multiple mobile

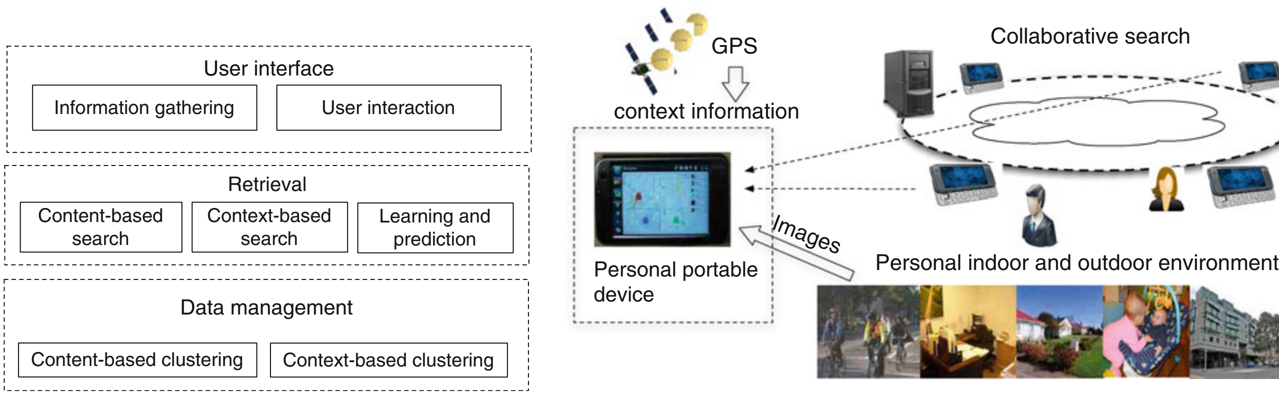

Fig. 1 System architecture overview of iScope: personalized multi-modality image search for mobile devices

devices, which enable efficient information sharing and effective social interaction in mobile social networks (Sect. 6).

iScope supports iterative personal image search on mobile devices. Personal images are organized using the hierarchical clustering structures of content and context on a local mobile device or multiple distributed devices. At each retrieval step, given a user's feedback, e.g., a query image or specific context information, iScope traverses through the hierarchical multi-modality data clusters stored either locally or remotely, predicts and identifies a potential match, and returns the candidate thumbnail images of the matched cluster back to the user. The search process continues until the target image(s) are found. Figure 2 illustrates the interactive search process. A user looks for a photograph taken during a hiking trip. Starting with a photograph of his recent paintball trip, the user conducts three context search and one content search operations. The first two photographs belong to the paintball trip (similar location); the last three photographs belong to the hiking trip (similar location, content); and the paintball trip occurred a week after the hiking trip (temporally similar).

3 Multi-modality data management

iScope uses a novel multi-modality image management scheme that supports both content and context (e.g., time, location, and ownership) information associated with images. This section explains how these data are obtained and used in multi-dimensional, multi-scale image clustering in oder to support efficient browsing and run-time user queries.

3.1 Incremental hierarchical clustering

iScope organizes multi-modality data using incremental hierarchical clustering [5], which supports order insensitive and efficient traversal of a data set across different dimensions and resolutions. Hierarchical clustering is an effective method in data analysis and interactive information retrieval. It yields a hierarchical clustering tree structure with multiple levels of resolution instead of a single-layer clustering structure. As such, it enables compact data management and efficient search. However, traditional hierarchical clustering techniques have a major shortcoming when dealing with incremental data growth. As users continue to add new images (newly captured, from friends, or from the Web) to their personal mobile devices or modify/remove certain images, periodical re-clustering may be needed—each time a complete hierarchical clustering process is performed on all images currently in a data set—introducing significant computation overhead and energy consumption to personal mobile platforms.

The inefficiency mentioned above is the result of neglecting previously constructed hierarchical clustering trees in traditional approaches and reconstructing a

Fig. 2 Image search example using context and content in iScope. Paintball trip (first 2 images) and hiking trip (last 3 images)

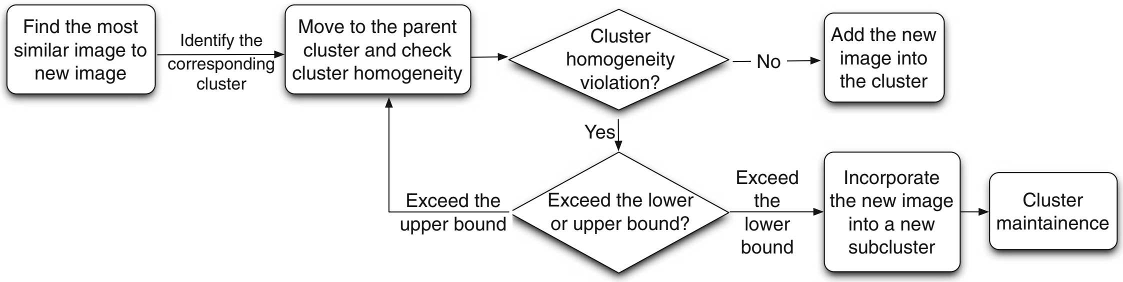

Fig. 3 Incremental hierarchical clustering: flow chart shows the key steps when a new image is inserted

clustering tree from scratch each time. In the incremental hierarchical clustering approach used by iScope, two intrinsic properties of a clustering tree structure are considered: *homogeneity* and *monotonicity*. Each cluster can be represented by its density, a triple $X = \{D, \mu, \sigma\}$, where $D = \{d_i | d_i \in \Re\}$ is a population of nearest distances between data items in the cluster, and μ and σ are the mean and the standard deviation of D, respectively. Given a lower bound $L = \mu - \sigma$ and an upper bound $U = \mu + \sigma$, a cluster is homogeneous if and only if $L \leq d_i \leq U, \forall\ d_i \in D$. For a homogeneous cluster of images, all the images have similar nearest neighbor distance. A cluster that violates homogeneity needs to be restructured. A cluster satisfies the monotonicity property if its density is always higher (i.e., images are more similar) than the density of its parent cluster.

In iScope, the clustering operations can be divided into two major procedures: tree construction and tree maintenance. Figure 3 illustrates the key steps when a new image is added. In the tree construction stage, the new image is added into the clustering hierarchy in a bottom-up fashion. Our algorithm first finds the nearest (most similar) image at the bottom level and recursively checks whether the parent cluster can host the new image with minimal density changes and minimal disruption to the hierarchy monotonicity. If cluster homogeneity is not violated, the new image is added to the cluster. Otherwise, the new image is added to a sub-cluster or a higher-level cluster depending on whether the lower bound or the upper bound is exceeded. Note that in this stage, only the regions affected by the addition of the new image will be restructured if necessary. In the tree maintenance stage, all affected clusters will be checked iteratively for homogeneity violations and restructured until all nearest distances of a cluster are within its lower and upper bounds.

Compared with traditional approaches, incremental hierarchical clustering avoids global data re-clustering and updates data much more efficiently. In addition, for small clusters with few images each, our clustering algorithm recursively merges two closest clusters if the merged cluster has at most M images, where M is the number of thumbnail images that can be displayed on the screen of a portable device (M is set to 24 in our study). Therefore, our incremental hierarchical clustering results are stable, insensitive to the order of data changes, and highly similar to the clustering results of traditional approaches (0.96 similarity on average, with 0.03 standard deviation for our data sets). Evaluation results of incremental hierarchical clustering will be elaborated in Sect. 7.

3.2 Content and context-based clustering

Given the raw content of an image, various types of image features may be extracted. We compared a number of features and chose to use a region-based feature called *region-abc*, which is similar to the feature used previously [6]. Here, each image is segmented into multiple coherent regions, and a 14-dimensional feature vector (9-dimensional color moments and 5-dimensional bounding box) is extracted from each region. The distance between two region feature vectors is defined as the sum of the best-matched distances for each individual region.

Using the region-abc image features, we apply the incremental hierarchical clustering algorithm to yield multiple clusters. On top of these clusters, we also construct a content-based cluster relationship graph in which each node represents a content cluster and each edge represents the distance between the centroids of two clusters. Using this graph, we can quickly identify other clusters that are likely to contain images with similar content to those in a given cluster, thus permitting efficient content-based image browsing and retrieval.

In addition to image content information, our system also uses various types of context metadata to improve image data management quality and efficiency.

Geographical location information of images can be captured by mobile devices equipped with GPS receivers, permitting easy computation of the spatial correlation among images. Figure 4 shows an example geographical distribution of a user's image data set. Unlike the content-based clustering technique described above (which

Fig. 4 Geographical distribution of a user's images

maintains a set of flat clusters), context-based clustering maintains the entire cluster hierarchy.

Similarly, hierarchical time clusters can be constructed, each containing images captured within a certain time period. Temporal correlation between images has been observed in many scenarios and can help identify images of certain activities (e.g., wedding) or images taken at a certain time (e.g., Macy's Thanksgiving Day Parade). In this work, distance in the time domain is measured as the absolute time difference between pairs of images. For distributed image sharing, ownership information also plays an important role. A user may obtain images from other users, and through the ownership information, identify other users with similar interest (e.g., classic cars). This can be used to restrict image browsing or searching to a specific set of friends, enabling more efficient image search and effective social interaction.

3.3 Interactions of multi-modality clusters

Figure 5 illustrates the iScope multi-modality image clustering method, using metadata information such as content, location, and time. At the bottom of the figure, images are clustered based on their content similarity. There are also links between content clusters indicating the closeness of cluster centroids. At the top of this figure, hierarchical geographical clusters (solid-line ellipses) and time clusters (dotted-line ellipses) are maintained at different resolutions, reflecting spatial or temporal correlations between images. Images belonging to the same content cluster may reside in different geographical or time clusters, and vice versa. For instance, a user may take a set of similar (or dissimilar) pictures at the same location during a certain period of time. Or, a user may have taken a lot of pictures of her dog at various time and locations. As a result, using

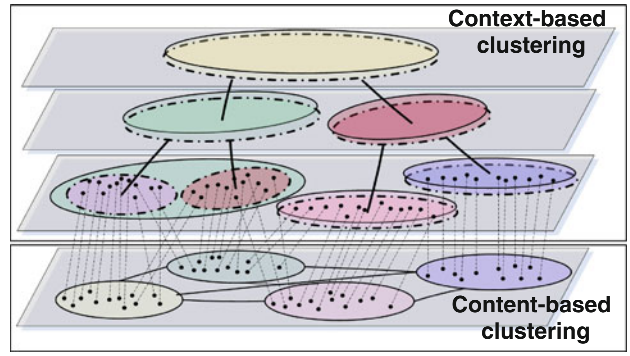

Fig. 5 Interactions of content and context-based image clusters

inter-connected multi-modality clusters makes it easier to capture higher-level image semantics. A user can quickly navigate these clusters by following different types of correlations (similar content, location, time, or ownership) in order to locate the images of interest. In addition, through adaptive user prediction (Sect. 5), iScope may automatically determine the most promising correlation without explicit user specification.

Clustering large amounts of image data using different types of metadata can be time consuming and memory intensive. To improve efficiency, a hybrid approach is used in which expensive clustering computation is performed on wall-powered server machines when a mobile device synchronizes with a server, and incrementally updated clusters are stored on mobile devices for efficient personalized image management and search.

4 Mobile platform characterization

In this section, we characterize the performance and energy use of image search in personal mobile systems.

4.1 Measurement setup

The measurement platform includes a Nokia N810 portable Internet tablet, HP Harrison 6201B direct current power supply, NI-PC-6034E acquisition card, and hosting workstation. iScope has been prototyped on Nokia N810, which is representative of modern personal mobile networked multimedia embedded systems. In particular, N810's 4.3 in LCD touch screen allows the design and evaluation of user-interactive search techniques for personal mobile devices. To measure energy and power consumption, we replace the battery of the mobile platform under test with an HP Harrison 6201B direct current power supply. Current is computed by measuring the voltage across a 5 W, 250 mΩ, Ohmite Lo-Mite 15FR025 molded silicone wire element resistor in series with the power supply. This resistor was

Table 1 Power consumption (W)

Processor active	Processor idle	Display w/o touch	Display w/ touch	Wireless send/receive
0.80	0.01	0.47	1.04	2.00/1.76

designed for current sensing applications. High-frequency voltage samples are taken using a National Instruments 6034E data acquisition board attached to the PCI bus of a host workstation. The board has a maximum sampling rate of 200,000 samples per second, allowing for high-resolution power and energy analysis of the mobile system.

4.2 Hardware power characterization

Next, we measure the power consumptions of the major components of the Nokia N810, including the TI OMAP embedded microprocessor, LCD touch screen, and Wi-Fi interface. The power consumptions of individual components are measured independently from others using specifically designed testing programs, and the testing environment is carefully controlled, so that interference from other components is eliminated. For instance, the screen is turned off when the processor is tested. The results are shown in Table 1. The peak (idle) power consumption of the microprocessor is 0.80 W (0.01 W), where peak power is measured when the microprocessor is doing intensive computation, such as image feature calculation. The power consumption of the touch screen is 1.04 and 0.47 W with and without being touched, respectively. The send (receive) power consumption of the wireless interface is 2.00 W (1.76 W). This study shows that the power consumption of the display is comparable to that of the microprocessor and wireless interface. This observation is critical in a user-interactive search process, in which the search system iteratively refines its search results based on user feedback until a satisfactory image is found. During the interactive search process, the energy consumption of human–machine interface components, e.g., the LCD touch screen, can be significant. In addition, the energy consumption of the wireless interface must be carefully considered during distributed collaborative image search and sharing among multiple mobile devices.

4.3 Image retrieval characterization

We now characterize the performance and energy consumption of the image search process. This study helps clarify the time breakdown and energy consumption distribution among the various steps of the image search process. Given an initial query image, users look for a target image using content-based search algorithms through an interactive search process. Image data set includes approximately 2,000 images taken on Nokia N810 portable Internet tablets.

Table 2 Time distribution of one image search process

	Seconds
Total time	80.4
Query dialog	8.4
Query idle	8.0
Query click	0.4
Algorithm computing	15.0
User exploration	57.0
Screen idle	53.3
Screen click	3.7

Table 3 Power distribution of one image search process

	Joule
Total energy	52.2
Query dialog	4.2
Query idle	3.8
Query click	0.4
Algorithm computing	19.1
User exploration	28.9
Screen idle	25.0
Screen click	3.9

Tables 2 and 3 show the time breakdown and energy consumption distribution of the search process, which has the following components: (1) the initialization stage, including user interface initialization and query image selection; (2) online processing of the content-based search algorithm, including inter-image similarity calculation; and (3) user exploration, including browsing, thinking, and selection. The measured time and energy breakdown among these three components are 10.5–18.7–70.9 and 8.0–36.6–55.4 %, respectively. Note that, in this study, image similarity is calculated at run time, which can also be conducted offline. Therefore, the user exploration stage dominates in both latency and energy consumption. This study demonstrates that personal image management and search should focus on minimizing the latency and therefore energy consumed in the user exploration stage. To this end, iScope employs multi-modality data management and user-centric adaptive search algorithms, which are explained in Sects. 5 and 6.

5 User-centric adaptive image search

This section describes the proposed user-centric image search techniques which leverage content and context information, as well as online adaptive user prediction during image search.

5.1 User interface

One of the main difficulties standing in the way of greater benefit from any intelligent search algorithm is difficulty of use. Most existing browsing-based user interfaces, although inefficient, are straightforward to use. iScope aims to make mobile image search accessible to large population of mobile system users spanning different age groups with different interests and technical backgrounds: an easy-to-use interface is essential.

We have designed a user interface that is accessible to people with no technical background. It supports queries via a straightforward search process. Figure 6 shows the prototype user interface implemented on a Nokia N810 device. The figure on the left shows the starting page, which shows the list of the social group members and the geographical distribution of the image data set. Two types of navigation are supported: (1) navigation across different dimensions, e.g., time, location, content, and ownership, corresponding to the search algorithm's traversal across different dimensions of metadata clustering and (2) zooming in/out along a particular dimension, corresponding to search traversal along a cluster hierarchy. Using this interface, an end user can conduct image search through an interactive navigation process. For instance, using a query image of a person running in Boulder, a user can search for a stadium in Toronto. First, content-based search is used to look for photographs with people running. Then, location-based search is used by selecting Toronto on the map to reduce the candidate data set. Manual browsing is then used to find one candidate image containing running people in a stadium. Finally, content-based search is used to search for stadiums in Toronto.

5.2 Search process

To search for an image, a user starts with an existing query image, related context information, or browses in an initial cluster to identify a specific query image. The user then selects a search domain (e.g., content, location, or time) and issues a query. Given the initial query, iScope quickly locates the corresponding cluster that contains the query image in that domain. As described in Sect. 3, images assigned to the same cluster are similar in a particular domain. Promising images can be easily identified by returning other images residing in the query image's cluster. These temporary results are presented to the user, who checks the images' context information and provides feedback on whether they are relevant. The user can then continue the interactive search in two different ways. The user may stay in the same search domain, and check the upper-level cluster (for geographical or time clustering) or the neighboring clusters (for content-based clustering). Alternatively, the user may pick one of the positive examples as the new query image and start another query, switching to another search domain if needed. This iterative search process continues until the desired image is located.

All the search steps and user feedback are recorded by iScope and used to tune the automatically generated clustering structures as follows: (a) if an image is selected as the target image or an intermediate target image, it is merged into the same cluster as the query image; (b) if a cluster contains more images than that can be displayed on the touch screen, the most irrelevant images will be identified and removed from the original cluster and form a new cluster; and (c) an empty cluster will be removed from the clustering structure.

5.3 Adaptive user prediction

In addition to explicit user feedback on relevant or irrelevant images, other types of implicit user feedback may also

Fig. 6 User interface running on Nokia N810. The figure on the *left* shows the start page, and the figure on the *right* shows search results. Last row in search results is based on adaptive prediction

be captured, such as the overall search and navigation path, backtracking operations. This information can be used to guide the run-time learning techniques and provide adaptive user prediction to optimize the user search process. Specifically, iScope makes user-specific prediction based on previous search history, current query image, and intermediate search results, in order to return images that are likely to be of interest.

Our method works as follows. After each round of search, the system records the trace $(q, h_1, h_2, \ldots, h_x, p)$, in which q is the initial query image, $h_1, h_2, \ldots, h_x$ are the intermediate images, and p is the final target image. This image-level trace is then converted to a cluster-level trace, i.e., each image is converted to its corresponding cluster and search domain. Cluster-level traces, instead of image-level traces, are used for prediction because users are unlikely to search for the same image repeatedly, but are likely to search for different images in a cluster (e.g., a specific event or a trip). Given a set of cluster-level traces, at runtime, iScope uses the images selected by the user so far in this round of search as a basis for prediction. Let $(i_1 i_2 i_3)$ be the corresponding clusters. Using Bayes' theorem, we calculate the conditional probability of each candidate cluster C containing the target image:

$$P(C|i_1 i_2 i_3) = \frac{P(i_1 i_2 i_3|C)P(C)}{P(i_i i_2 i_3)} \quad (1)$$

Since $P(i_1 i_2 i_3)$ is the same for all candidate clusters C, we only need to compute $P(i_1 i_2 i_3 \mid C)P(C)$. Again, using Bayes' theorem, we have

$$P(i_1 i_2 i_3|C) = P(i_1|C)P(i_2|i_1 C)P(i_3|i_1 i_2 C) \quad (2)$$

Using the naive Bayes probabilistic model, i.e., $i_1 i_2 i_3$ are conditionally independent of each other, we have

$$P(i_1 i_2 i_3|C) = P(i_1|C)P(i_2|C)P(i_3|C) \quad (3)$$

We first locate all the cluster-level traces that contain C, then check how many times i_1, i_2, and i_3 have co-occurred with C in these traces. To compute $P(C)$, we count the number of occurrences of C in all the cluster-level traces O_C, and the total number of cluster occurrences in the traces O. Thus,

$$P(C) = O_C/O \quad (4)$$

Using the formulas above, we can compute the probability of each candidate cluster containing the target image.

6 Collaborative image search

iScope supports collaborative image search targeting distributed mobile environments. The proposed design allows individual users to share their image data sets within their social groups, e.g., friends and family members. It thus allows each member to search a much larger data set than a single mobile device can hold, thereby facilitating information sharing and stimulating social interaction. Previous work has shown that collaborative search utilizing social networks (e.g., friends or social groups) can improve search efficiency and generate more relevant search results [7–9]. While previous work focused mostly on keyword-based search of Web data, iScope focuses on collaborative content- and context-based search in distributed mobile systems. Privacy and security are key issues in data sharing. iScope can leverage existing infrastructures for authentication and privacy/data protection [10, 11].

The collaborative search technique conducts parallel search among the socially associated mobile devices. A search query may be processed by multiple mobile devices, each of which hosts a different image data set manually shared by its owner or automatically cached by the device itself. Each member of a social group shares a subset of her image data set with the whole group. The shared data set is initially stored on her own device and organized separately from the rest of her personal data. More specifically, the metadata management using the multi-modality clustering method is separated from the rest of the owner's personal data for better privacy and security protection. This approach yields smaller data set, potentially allowing more efficient image search. The distributed shared data sets are ready to support collaborative image search within a social group. After a group member issues a query, her local device conducts local search within her own data set. In the meantime, the query is broadcasted to other devices within the social group. Each remote device will collaboratively conduct local search within its shared data set and return the results, e.g., metadata and/or the raw images, back to the querying device. The user interface for distributed search is identical to that of local search; the remote search process is transparent to the end user. Figure 7 demonstrates the flow of collaborative search.

In this work, we descrive an online metadata caching method to minimize the communication overhead of collaborative search. We observed that individual users tend to show more interest in specific subsets of the shared data, and the subsets of interest vary between users. For instance, Alice and her friend Bob took a hiking trip to the Greenman Summit. Alice may be more interested in the photographs taken by Bob during the trip than Bob's other shared data. The proposed caching method leverages the "data locality" property and caches the metadata received remotely at run time, merging the metadata into the user's own data set for future usage. In addition, to support collaborative search, image ownership is introduced as a dimension in the multi-modality data clustering method. When local search requires access to a remote image, it first checks metadata referencing remote storage and then

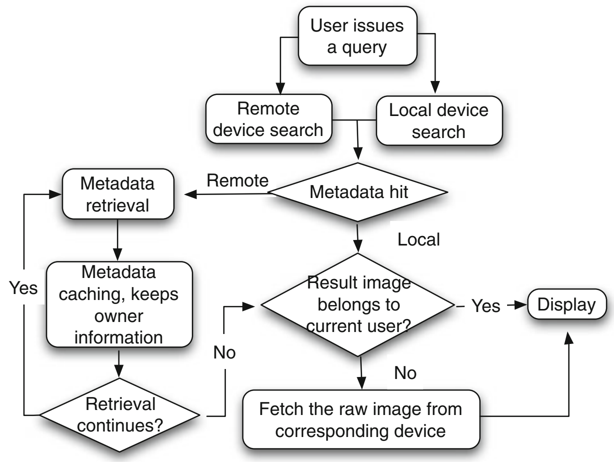

Fig. 7 Collaborative search flow chart

issues a fetch request to the corresponding device. If that device is currently available in the network, it returns the raw image requested.

As described in the previous section, metadata clustering is hierarchical. The proposed metadata caching method follows a bottom-up approach, i.e., when all the sub-clusters of a remote cluster have been cached locally, the remote cluster itself is then cached. In addition, each cached remote metadata item and the corresponding cluster also maintain an access history, which tracks how many times, and the most recent time at which, the corresponding image(s) have been accessed. This information is used to determine the caching policy for the raw images, which are much larger than metadata. When a device has insufficient storage, the raw images with low accesses counts, or long durations since their most recent access, are deleted.

7 Experimental evaluation

In this section, we evaluate iScope, the personalized image management and search system. Section 7.1 summarizes the implementation of our prototype and describes the image data sets used in the experiments. Section 7.2 evaluates multi-modality data clustering algorithm. Section 7.3 evaluates personalized image search on an individual device. Section 7.4 further analyzes users' search patterns and importance of cue images in online prediction. Section 7.5 evaluates collaborative search in a distributed mobile environment.

7.1 Implementation and image data sets

iScope has been implemented on a Nokia N810 device. The multi-modality image data management method, as well as content-based and context-based search techniques are implemented in C and Python. The GTK+ library was used to develop the graphical user interface. The implementation consists of 23,925 lines of C code and 669 lines of Python code.

Sets of images captured using personal portable devices, such as camera phones, are significantly different from general-purpose image data sets. We have constructed an image data set with 7,923 Flickr images captured by six different camera phone users. The Flickr data set is used in the evaluation of content-based search techniques in Sect. 7.2, because it is more comprehensive (requiring that user study participants gather 8,000 images each would be costly), and this evaluation does not require any context information. However, these Flickr images lack personal context information, such as location and time stamps. In order to evaluate the impact of this context metadata, it was necessary to gather our own image data sets. We developed a software tool for Nokia N810 portable Internet tablets that allows users to manually or automatically take photographs using the built-in camera. The software uses the built-in GPS device and clock to tag photographs with location tags and timestamps. Ten volunteers took photographs during their daily activities. In total, they gathered more than 9,000 images during a period of four months. The images were taken in seven cities of three different countries: Canada (Kingston, Ottawa, and Toronto), the United States (Evanston, Boulder, and San Jose), and the United Kingdom (London). The gathered image data sets, along with the location, time, and ownership information, are stored on N810 devices. They are used to evaluate the impact of distance measurement on content and context clustering quality, as shown in Sect. 7.2 and in the user study shown in Sects. 7.3 and 7.5.

7.2 Multi-modality data management

iScope combines both content-based image features and context metadata to support efficient image data management. We first compare the efficiency of incremental hierarchical clustering (IHC) and traditional hierarchical clustering (THC). We select three subsets containing 100, 1,000, and 5,000 images, respectively. Table 4 shows the total clustering time by each algorithm as images are continuously added to the system. According to the results,

Table 4 Clustering time comparison of traditional (THC) and incremental hierarchical clustering (IHC)

Number of images	100	1,000	5,000
THC	0.341 s	366.5 s	43965.6 s
IHC	0.006 s	0.924 s	33.0 s
Ratio	62.0	396.9	1,334.2

the incremental approach outperforms the traditional approach by orders of magnitude, and the improvement becomes more substantial as the number of images increases.

Next, we compare the quality of IHC and THC. This evaluation is conducted on the data set containing 5000 images, since smaller data sets result in fewer clusters and do not reflect the overall quality. In THC, there are three distance measures by which sub-clusters are merged. We experimented with all of them: min, max, and avg, which measure the minimum, maximum, and average distances of objects belonging to two different clusters, respectively. A good clustering algorithm should generate clusters that are compact (small intra-cluster distance) and have good separation (large inter-cluster distance). Given a set of k clusters $X_1, X_2, \ldots, X_k$, we define the average intra- and inter-cluster distances as follows:

$$\mathrm{IntraDist} = \frac{1}{k} \sum_{1 \le i \le k} \left(\frac{1}{|X_i|} \sum_{x \in X_i} \mathrm{dist}(x, \overline{X}_i) \right), \tag{5}$$

$$\mathrm{InterDist} = \frac{2}{k(k-1)} \sum_{1 \le i \le j \le k} \mathrm{dist}(\overline{X}_i, \overline{X}_j) \tag{6}$$

where $\overline{X}_i$ is the centroid of cluster X_i and $dist()$ measures the distance between two objects. Table 7 shows the quality of different clustering algorithms. A higher *InterDist* to *IntraDist* ratio indicates better separation and compactness, i.e., better clustering quality. According to the ratio, the incremental approach results in slightly worse results. This small gap is acceptable given the much higher efficiency of IHC. We have chosen to use IHC for clustering content- and context-based information for mobile data management, due to its high clustering efficiency and good clustering quality.

7.3 Personalized image search

Here, we evaluate the personalized local image search system on the Nokia N810 platform via user studies. Ten volunteers from Queen's University and University of Colorado participated in the studies. All ten participants are graduate students aged between 20 and 28 years. Two of them are female. Most of the participants use mobile devices daily and have at least basic computer skills. We compare the performance of different image search algorithms under two different scenarios: (1) search within individual users' own image data sets and (2) search within a large combined image data set. Specifically, we performed two user studies, each with five participants. The amount of time spent by each participant ranged from 4 to 8 hours.

In the first study, users search for images within their own image data sets (Sect. 7.1). These data sets contain 1,079, 1,235, 1,497, 1,542, and 2,100 images, respectively. The data sets differ in size as different users collect images at different rates and under different scenarios. Although the data sets are different, all the five participants follow the same search protocol. In the second study, we use a larger image data set containing 4,389 images, which are drawn from three participants' data sets in the first study. Users are asked to familiarize themselves with other users' images in this data set, to minimize any affect caused by unfamiliarity of a user to the data set.

For each participant, 30 query images are randomly selected from the corresponding image data set, and the 30 target images corresponding to the query images, respectively, are then manually specified. Although it is possible to select multiple target images for each query image, using query–target image pairs provide a simple and more deterministic evaluation process. Figure 8 shows the clusters (based on content, location, and time) that the query images belong to in one of the image data sets. We see that the query images are distributed evenly and sparsely in different clusters in the three dimensions, thus ensuring that the user study results are not biased.

To evaluate the effectiveness of the design, we consider the following search scenarios:

- *Browsing-based search:* To date, browsing is the most commonly used search method for personal image collections stored on commercial mobile platforms. In this experiment, images are sorted by time. Given a query image, the user searches for the target image by browsing through the image data set.
- *Clustering-based search:* This approach leverages the multi-modality clustering data structures, content- and context-based search techniques. The described adaptive user prediction technique is disabled in this setting.

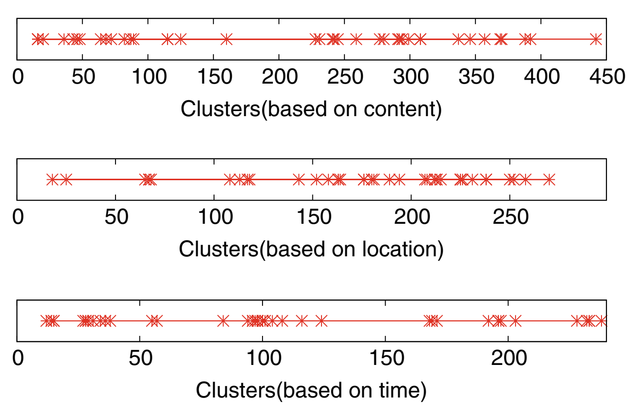

Fig. 8 Distribution of query images over the content, location, and time clusters. The x axis represents the individual leaf clusters in the content, location, and time hierarchical clustering trees, and the points show the clusters that the query images belong to

Table 5 Time usage of browsing-based search

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
Computation time (s)	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
User time (s)	130.5	51.3	41.0	120.9	125.2	449.4	137.1	141.1	595.7	219.0
Overall time (s)	130.8	51.7	41.3	121.3	125.6	449.8	137.5	141.5	596.0	219.4
Avg. steps per query	31.4	40.1	20.2	18.8	86.9	104.7	101.7	111.7	112.7	107.7

Table 6 Energy usage of browsing-based search

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
Computation energy (J)	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
User energy (J)	68.7	26.3	20.8	58.1	63.6	216.2	69.4	70.9	292.9	114.4
Overall energy (J)	69.1	26.7	21.2	58.5	64.0	216.6	69.9	71.3	293.4	114.9

- *Clustering+Prediction-based search:* This is the method used in iScope. In addition to clustering-based search, it also leverages implicit user feedback information from previous search history for user-specific prediction.

In this study, the system is configured to display 24 (4×6) thumbnail images at a time on the N810 touch screen. With adaptive prediction, the two images most frequently used in the image-level traces are selected from each of the top three clusters. These six suggested images are presented as the bottom row in the search results (see Fig. 6).

Tables 5 and 6 show the overall performance and energy consumption, as well as the time and energy usage breakdown, of browsing-based search. As described in Sect. 4, the time and energy usage of an image search process can be divided into two components: algorithm processing (Computation) and user operation (User). Using manual browsing, the time and energy overhead of the search algorithm (image index computation) is negligible. User operations dominate the search process. On average, more than 99 % of the time and energy is consumed by user operation (manual browsing). Table 5 also shows the average number of steps required by each user per image search. The manual browsing-based search process is tedious and slow (on average >100 steps per image for each of the five large image sets), resulting in significant time and energy overhead. We conclude that in the image search process, user interaction is the most time and energy consuming stage. Therefore, minimizing the number of required search steps has the greatest potential to minimize the time and energy usage (Table 7).

Figures 9 and 10 compare the time and energy usage of the browsing-based method, clustering-based method, and clustering+prediction (iScope). Compared to the browsing-based method, clustering-based search reduces search

Table 7 Clustering quality comparison of traditional (THC) and incremental hierarchical clustering (IHC)

	THC			IHC
	Min	Max	Average	
InterDist	720.3×10^6	549.8×10^6	750.7×10^6	11.57×10^6
IntraDist	81	100	67	7.18
Ratio	8.89×10^6	5.50×10^6	11.20×10^6	1.61×10^6

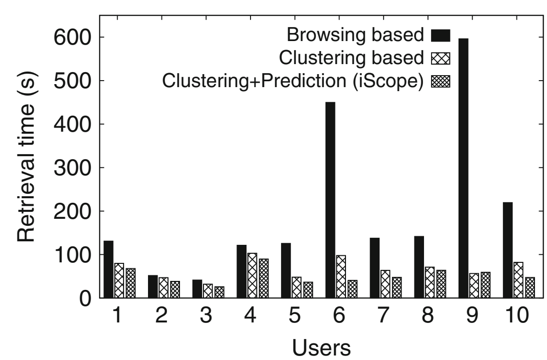

Fig. 9 Time comparison of search techniques

time and energy usage by 48.3 and 46.2 % (on average), 9.3 and 10.3 % (minimum), and 90.5 and 90.0 % (maximum). Leveraging the proposed adaptive user prediction technique, iScope further reduces the search time and energy usage by another 22.1 and 21.6 % on average, compared to the clustering-based approach. Overall, compared to the browsing-based approach, iScope achieves performance improvements of $4.1\times$ (on average), $1.3\times$ (minimum), and $11.1\times$ (maximum). It reduces energy consumption by $3.8\times$ (on average), $1.3\times$ (minimum), and $10.4\times$ (maximum). These experiments also suggest that the benefits of iScope increase when it is used on larger data sets. It enabled $1.9\times$ latency reduction and

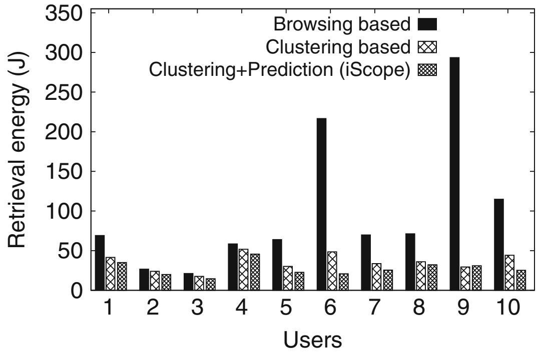

Fig. 10 Energy comparison of search techniques

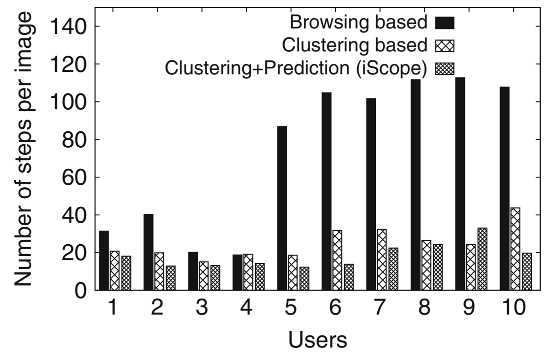

Fig. 11 Average number of search per query image

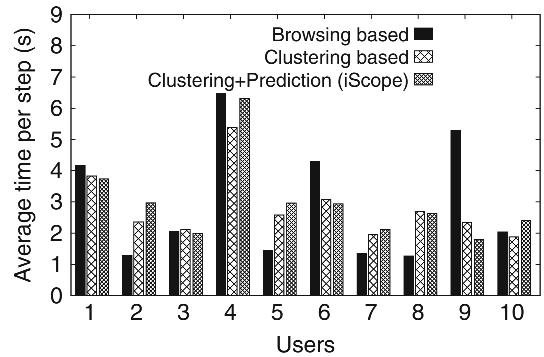

Fig. 12 Average time usage per search step

2.0× energy reduction when used for a 1,079 image data set, and 11.1× latency reduction and 10.4× energy reduction when used for a 4,389 image data set. Note that the user studies were conducted on different volunteers, and the content of different image data sets also varies significantly.

Figures 11 and 12 show the required number of search steps and the average duration of each search step for the three search techniques. In order to evaluate how different query images affect the user search process, we calculate the standard deviation of number of search steps for each user's 30 query images, which ranges from 10.5 to 83.9. The performance improvements and energy savings of iScope are primarily due to the significant reduction in the required number of steps for each image search query. In order to estimate the statistical confidence in our hypotheses about the impact of search algorithm on time and energy, we use the two-tailed Student's *t* test. The results of this analysis imply that the mean times for iScope and browsing mode differ with 97.3 % probability and that the mean energy consumptions differ with 97.0 % probability. Note that the *t* test requires some assumptions, e.g., that the variances of the two populations are equal.

The proposed multi-modality clustering and adaptive content and context-based searching techniques allow iScope users to use the implicit connections between the query and target images, thereby improving search quality and time. Consider the search processes shown in Fig. 13. In this case, the query image shows User 3's apartment in Kingston, and the target image (in User 3's image set) shows User 5's apartment in Boulder. Starting from the query image, through context (location), content, and context (location) search operations, User 3 reached an image containing a business building in Boulder. At this point, one context (location) search followed by a predictive content search (done automatically by iScope) was sufficient to reach the desired image. Note that, in this case, even though the query image and the target image contain similar "content," i.e., apartment, using only content-based search would result in an excessively long search process due to the two images' significant differences in color scheme and background content.

This study raises an interesting research question. Many times, we have heard people complaining, "I have seen this somewhere, but just cannot remember where." Recent studies, such as the SenseCam project [12], have shown that using image recording to enable review of one's daily life can ameliorate human memory loss symptoms. iScope explicitly leverages underlying connections among images. Its use may therefore have the potential to help people strengthen these connections. Currently, we are in the process of evaluating the possibility of applying iScope to related medical applications.

7.4 Further analysis of adaptive prediction

We further analyze the search traces of all users to understand the specific scenarios when iScope's clustering+prediction method improves or worsens the search performance. When personalized adaptive prediction is used, iScope dynamically predicts and displays a set of images that are likely to be the target image. The user may or may not choose a predicted image, and the overall search performance may be better or worse, compared with the performance when no prediction is used. Specifically,

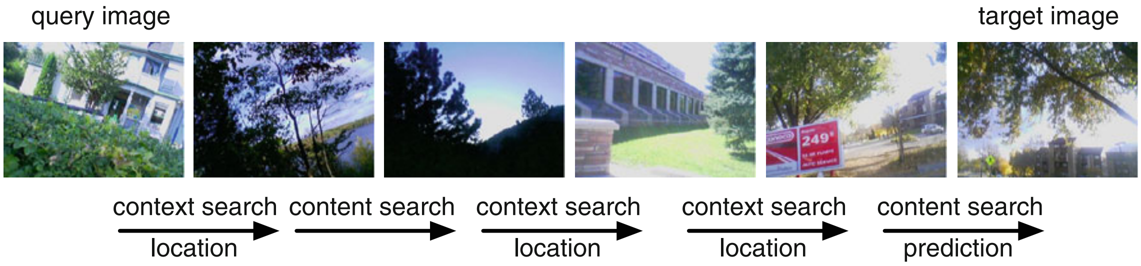

Fig. 13 An image search example using content- and context-based search and adaptive user prediction

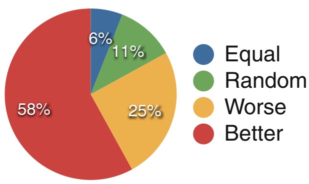

Fig. 14 Distribution of different cases of adaptive prediction

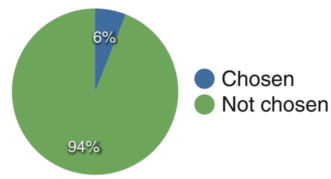

Fig. 15 Only 6 % of all images were chosen by users in their searches

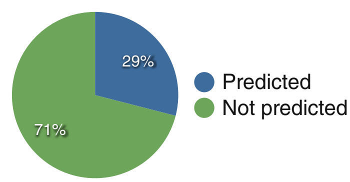

Fig. 16 Twenty-nine percentage of images chosen by users were predicted by iScope in the searches

we consider the following four scenarios: (1)–(3) if a predicted image is picked by the user, overall it may lead to *better*, the *same*, or *worse* search performance; and (4) if no predicted image is picked, the prediction's impact on the search performance is unknown, referred to as the *random* case. Figure 14 shows the distribution of the four scenarios in all users' search traces. Figures 17 and 18 show the corresponding changes (decrease or increase) of the number of search steps and search duration needed for the *better*, *worse*, or *random* cases.

From these figures, we can see that for most participants, using active prediction results in better search performance. For instance, *better* cases account for 58 % of all search traces, and these cases can reduce search duration by thousands of seconds and save hundreds of search steps.

Why does our personalized active prediction method work well in users' image search tasks? A careful analysis of users' search traces revealed an important property in users' image browsing activities: although many images exist in each user's data set and multiple candidate images are displayed in each search step, users tend to choose only a small portion of these images and use them repeatedly in their searches. For the 10 users, we have studied, on average, only 6 % of the images were actually chosen by our users (Fig. 15). Intuitively, this means that a user is often familiar with a small number of "hub" or "cue" images and use these images to make "jumps" in order to locate other related images. This is particularly true in personal image data sets.

Based on this observation, images used by a user in past searches have a greater chance to be used again in future searches, and these images that a user has revisited over and over again in the history should rank higher in the list of predicted images. In particular, iScope's personalized active prediction aims to extract such implicit information, as introduced in Sect. 5.3. Predictions are made based on previous search history, only those images used most frequently in the image-level traces will be selected and presented in the predicted image list. Figure 16 shows that 29 % of all images chosen by the participants were images predicted by iScope. Note that a random prediction method would only be able to predict 6 % of the images that were chosen (Fig. 15). Figure 19 presents the number of occurrences of images being chosen and being predicted. As we can see, the more frequently an image has been chosen, the more frequently this image appears in the predicted image list, thus the better prediction accuracy and better search performance (Figs. 17, 18).

The underlying reason for the success of iScope's personalized active prediction method is its close relationship with the way human memory works—when faced with a large amount of data, people tend to remember a few characteristic data items and use them to associate with other data items with similar features. In the case of personal image data management, users make use of a small number of "cue" images to facilitate browsing and searching of much larger image repositories. By organizing image data sets in multiple dimensions (e.g., content, time,

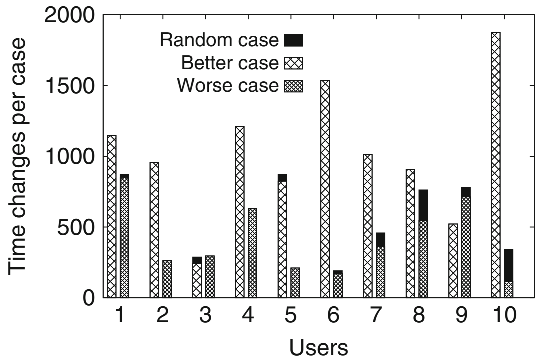

Fig. 17 Change of search duration using active prediction

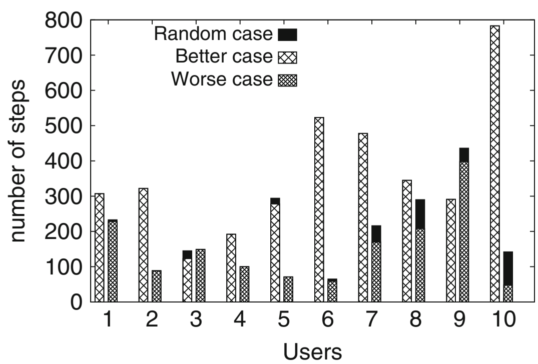

Fig. 18 Change of search steps using active prediction

location), actively learning, and predicting personalized search traces, iScope makes it possible to manage, search, and share large amounts personal image data in distributed mobile platforms with high efficiency.

7.5 Collaborative image search

The distributed, collaborative image search technique described in Sect. 6 was also evaluated. Communication latency and energy overhead are of primary concern in collaborative search. The caching technique aims to minimize these overheads by limiting remote access during collaborative search.

The following experiments consider N810 devices connected via a campus 802.11b network. The user studies described in the previous section were extended to the distributed environment. Detailed image search traces were gathered during the preceding local search experiments. These traces contain detailed timing information for the interactive image search processes, e.g., the number of search steps of each image query, the time usage of each search step and the breakdown between algorithm processing time and user time. The traces were replayed in the distributed, networked system composed of N810 devices. This technique has the benefits of eliminating ordering effects and random variation between the two studies. It also allows a more direct comparison of local search with distributed collaborative search than would be possible by repeating the study with a new set of users. Timing and system state information was gathered at run time. For instance, networking latency and energy consumption are gathered when remote device accesses are invoked. The power consumption of the N810 in each system state (e.g., receiving data via the 802.11b interface, running a search algorithm, and waiting for user input) was measured using the equipment described in Sect. 4. These system state-dependent power consumption values were used in combination with the timing and system state values measured during trace execution to determine the energy consumption during distributed collaborative search.

We first evaluate the potential communication performance and energy overhead introduced by remote access. In this experiment, the image data set is placed on remote devices and the proposed caching technique is disabled. Therefore, every image search step requires remote device access. Figures 20 and 21 show the energy usage and latency breakdown of the remote search scenario. Compared to image search on a local standalone device, remote image search introduces significant latency and energy overheads. The latency increases by 65.5 % on average (27.1 % minimum and 96.4 % maximum) for the ten participants in user studies. The corresponding total energy consumption increases by 607.5 % (275.5 % minimum to 877.7 % maximum), which includes the energy consumption of the querying device and the remote devices. Note that, since all the remote devices can potentially respond to each query, the worst-case latency and energy overhead increases linearly with the number of mobile devices (four devices are used in this experiment). This study illustrates the importance of reducing the communication overhead during distributed collaborative search.

Figures 22 and 23 compare the performance and energy usage of collaborative search with (right bars) and without (left bars) the metadata caching technique. These results

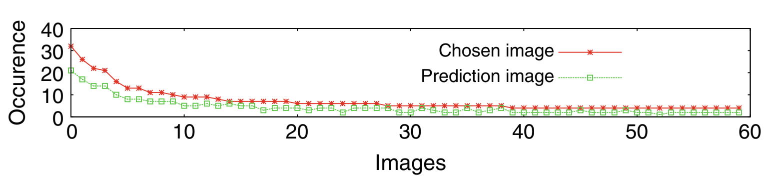

Fig. 19 Number of occurrences of images being chosen and being predicted. Images that are frequently chosen are predicted more frequently

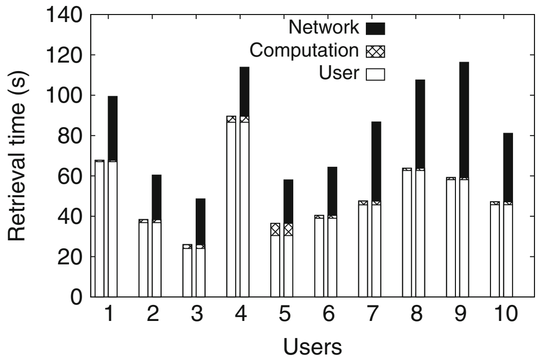

Fig. 20 Time breakdown, local (*left bars*) remote (*right bars*) retrieval

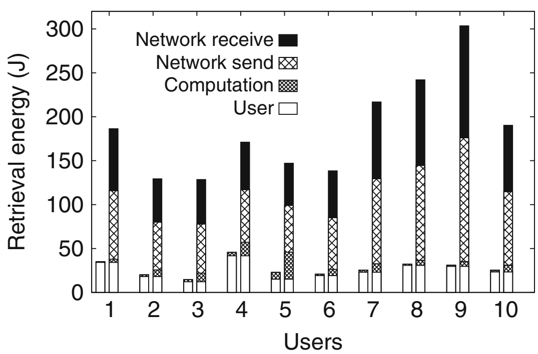

Fig. 21 Energy breakdown, local (*left bars*) remote (*right bars*) retrieval

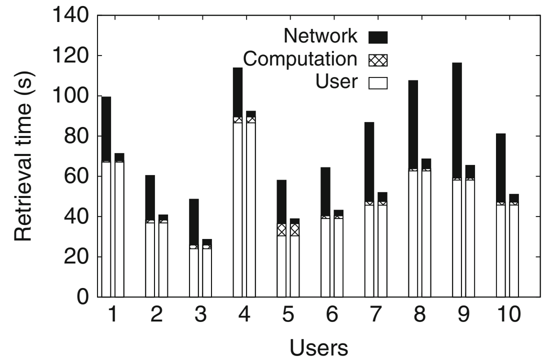

Fig. 22 Time breakdown, with (*right bars*) or without (*left bars*) caching

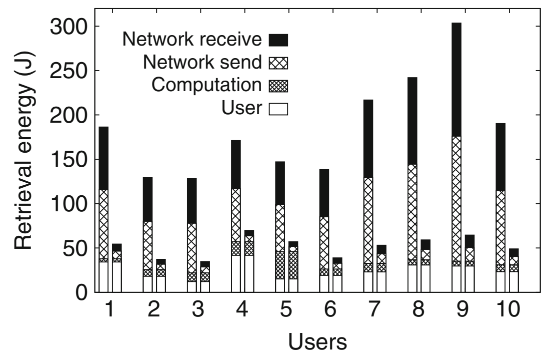

Fig. 23 Energy breakdown, with (*right bars*) or without (*left bars*) caching

demonstrate that metadata caching improves system performance and energy efficiency. When both five-member user studies are considered, the latency reduction is 34.4 % on average (18.9 % minimum and 43.7 % maximum) and the energy consumption reduction is 71.2 % on average (59.2 % minimum and 78.7 % maximum). These performance and energy consumption improvements result from high cache hit rates during the search processes. Table 8 shows the average cache hit rates over the user studies, which average 81 % and range from 69 to 90 %. This study also demonstrates that the cache hit rate decreases with increasing data set size—image search of a larger data set tends to be more diverse, lowering the cache hit rate. Note that the cache hit rate is affected by the query image distribution within the image data set. In practice, we believe that personalized queries generally have content and/or context correlation, which is reflected as data locality during image search, enabling a high cache hit rate. In contrast, the query images used in this experiment are randomly selected. Therefore, we believe iScope's caching techniques will be even more effective in real usage scenarios. Figure 24 shows the cache hit rate profiles of the ten participants in user studies; the cache hit rate increases for each participant—initially, the local device only contains its own data set and its cache is empty, resulting in a low cache hit rate. As queries are processed, more metadata are cached, improving the cache hit rate.

8 Related work

Our work draws upon research in several areas concerning image management: content-based image retrieval, multimodality image management, power-aware image retrieval, user feedback, and distributed image sharing. In this section, we survey work most related to ours.

Content-based image retrieval (CBIR) has been an active research area for over a decade [3]. Several approaches, aiming at providing a more intuitive interface for browsing and managing image collections, have been introduced [13, 14]. Content-based search for images taken by mobile devices has also been investigated [15, 16]. In this work, targeting personal image collections, we envision more active roles for portable devices and personalized search.

Besides the raw content of image data, researchers have also considered other types of information in order to augment image management and search tasks. Text

Table 8 Average cache hit rate for collaborative image search

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	Average
Cache hit rate (%)	90	85	84	86	79	69	75	81	85	76	81

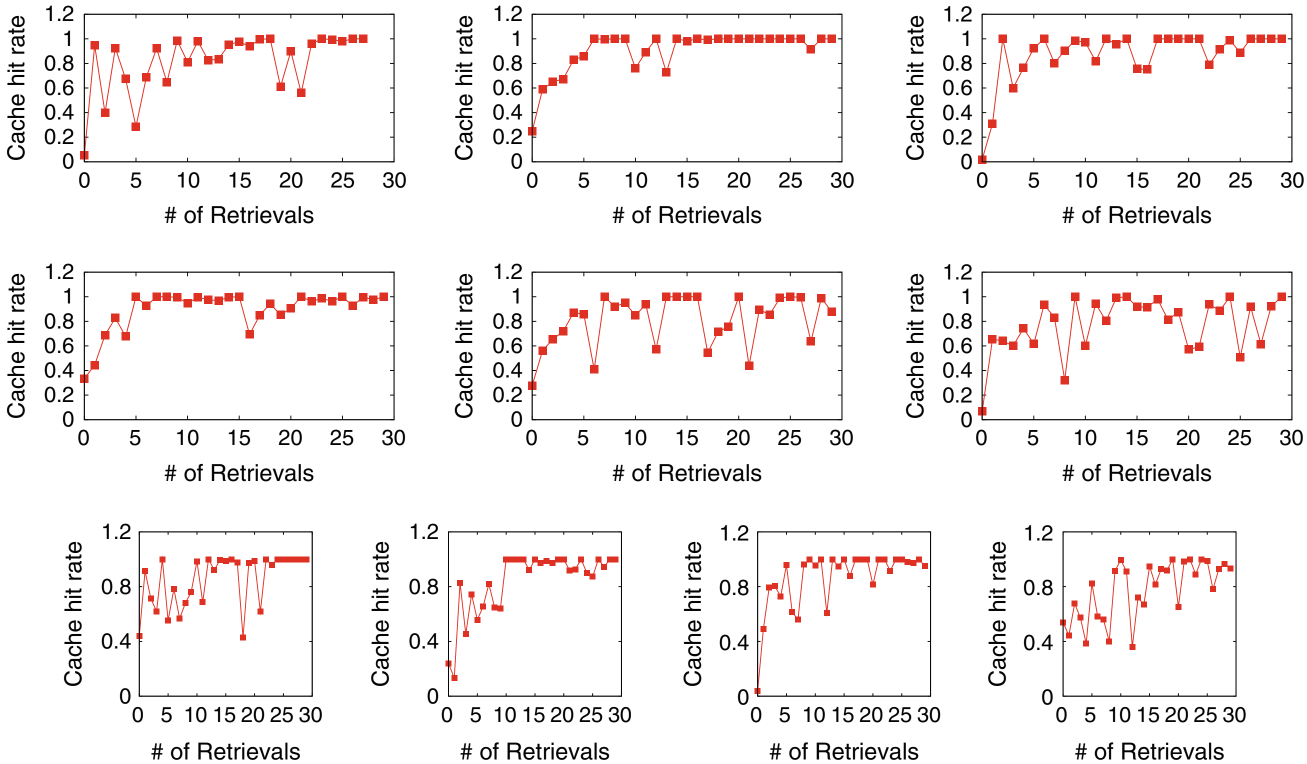

Fig. 24 Collaborative search: Cache hit rate profile for the ten users

annotations, Web links, and ontology have been considered in previous works.

For mobile images, location information is commonly used [17, 18]. Although these past works utilized context information, they did not carefully consider the energy issue, which is the primary constraint of battery-powered systems.

A variety of energy optimization techniques have been proposed for portable devices [19, 20].

Recently, Kumar et al. [21] proposed an adaptive feature loading scheme for mobile CBIR to save energy. This work focused on the energy consumption of CBIR processing. However, our study has shown that for image search on mobile devices, power consumption is mainly due to various components such as touch screen and GPS, instead of processor or storage.

Relevance feedback has attracted much attention in the information retrieval community and has been shown to provide improved performance in many search systems [22–24]. Most user feedback mechanisms aim at precision/recall improvement and ignore the speed issue, which is an important factor for performance measurement and power consumption in mobile systems. Saha et al. presented a human perception based similarity measure along with a relevance feedback indexing scheme [25]. Different from the past works, our study shows that, in many cases, the adjacent user search steps show little correlation. Therefore, we propose a naive Bayes' classifier-based algorithm for image prediction.

Distributed data sharing for mobile devices has been a popular research topic. Several general-purpose systems have been developed [26, 27]. A distributed image search scheme has been proposed by Yan et al. [28] for camera sensor networks; it does not target personal images. Other social-oriented multimedia and sensing data sharing systems include Micro-Blog [29] and CenceMe [30]. People's data sharing needs on mobile devices have also been studied [31, 32]. In our work, a metadata caching technique is proposed to effectively minimize the communication overhead during collaborative search.

Compared with the past works, our study shows that user interactions and communication dominate system energy consumption. iScope leverages both content and context information, as well as learning techniques, for personalized, energy-efficient image management, search, and sharing.

9 Conclusions and future work

In this work, we have described and evaluated iScope, a user-centric system for personal image data management, search, and sharing on mobile devices. iScope uses new techniques for multi-modality clustering of both content and context information for efficient image data management, as well as user-centric search algorithms with adaptive user prediction tailored to individual users. It also supports distributed image sharing and search with online metadata caching. We have implemented a prototype of iScope on networked Nokia N810 portable Internet tablets, and experimentally evaluated it via user studies. Our results show that, on average, iScope improves on the search speed and energy consumption of browsing by 4.1× and 3.8× , respectively. Also, the use of metadata caching in distributed image search reduces search latency by 34.4 % and reduces energy consumption by 71.2 %.

Our analysis of users' search traces also reveals the important relationship between the way human memory works and people's search patterns. Specifically, users tend to use a very small number of "cue" images to facilitate their search processes. These images contain certain characteristics that can provide the implicit cues for users to recall or recognize something easily. Identifying and utilizing such images are thus critical for large-scale information retrieval and management.

The future work includes exploration of more efficient parallel search algorithms to further minimize the communication overhead of collaborative search. In addition, we are interested in determining whether implicit multi-modality search techniques, such as iScope, have the potential to improve human memory or counteract memory loss. Finally, we will further investigate prediction algorithms to incorporate the sequential dependencies of user feedback during personal image search.

References

1. American museum. http://americanhistory.si.edu/
2. Measuring the information society (2011) International Telecommunications Union, Sept
3. Datta R, Joshi D, Li J, Wang JZ (2008) Image retrieval: ideas, influences, and trends of the new age. ACM Comput Surv 40(2):1–60
4. Zhu C, Li K, Lv Q, Shang L, Dick RP (2009) iScope: personalized multi-modality image search for mobile devices. In: MobiSys'09, pp 277–290
5. Widyantoro DH, Ioerger TR, Yen J (2002) An incremental approach to building a cluster hierarchy. In: ICDM '02: proceedings of the 2002 IEEE international conference on data mining
6. Lv Q, Charikar M, Li K, Kai (2004) Image similarity search with compact data structures. In: Proc. of the 13th ACM conf. on information and knowledge management, Nov
7. Dalal M (2007) Personalized social & real-time collaborative search. In: Proc. of the 16th intl. conf. on World Wide Web
8. Bao S, Xue G, Wu X, Yu Y, Fei B, Su Z (2007) Optimizing web search using social annotations. In: Proc. of the 16th intl. conf. on World Wide Web
9. Heymann P, Koutrika G, Garcia-Molina H (2008) Can social bookmarking improve web search? In: Proceedings of 1st ACM international conference on web search and data mining
10. Joels A (2006) RFID security and privacy: a research survey. IEEE J Sel Areas Commun (J-SAC) 24(2):381–395
11. Heikkila FM (2007) Encryption: security considerations for portable media devices. Secur Priv IEEE 5(4):22–27
12. Berry E, Kapur N, Williams L, Hodges S, Watson P, Smyth G, Srinivasan J, Smith R, Wilson B, Wood K The use of a wearable camera, sensecam, as a pictorial diary to improve autobiographical memory in a patient with limbic encephalitis: a preliminary report
13. Heesch D (2008) A survey of browsing models for content based image retrieval. Multimedia Tools Appl 40:261–284
14. Ardizzone MME, La Cascia M, Vella F (2010) Three-domain image representation for personal photo album management. In: Proc. SPIE 7540, 75400Y
15. Jia M, Fan X, Xie X, Li M, Ma W-Y (2006) Photo-to-search: using camera phones to inquire of the surrounding world. In: MDM '06: proceedings of the 7th international conference on mobile data management, p 46
16. Ahmad I, Abdullah S, Kiranyaz S, Gabbouj M (2005) Content-based image retrieval on mobile devices. In: Proc. of SPIE, vol 5684, Jan 2005
17. Mattias Rost HC, Holmquist L (2011) Mobile exploration of geotagged photographs. Pers Ubiquitous Comput 16(6):665–676
18. Anguera X, Xu J, Oliver N (2008) Multimodal photo annotation and retrieval on a mobile phone. In: Proceeding of the 1st ACM international conference on multimedia information retrieval
19. Chakraborty S, Dong Y, Yau DKY, Lui JC (2006) On the effectiveness of movement prediction to reduce energy consumption in wireless communication. IEEE Trans Mob Comput 5(2):157–169
20. Karagiannis T, Boudec J-YL, Vojnović M (2007) Power law and exponential decay of inter contact times between mobile devices. In: MobiCom'07, pp 183–194
21. Kumar K, Nimmagadda Y, Hong Y-J, Lu Y-H (2008) Energy conservation by adaptive feature loading for mobile content-based image retrieval. In: Proc. of the 13th intl. symposium on low power electronics and design
22. Hoi SCH, Lyu MR, Jin R (2006) A unified log-based relevance feedback scheme for image retrieval. IEEE Trans Knowl Data Eng 18:509–524
23. Liu W, Jiang W, Chang S-F (2008) Relevance aggregation projections for image retrieval. In: Proc. of the 2008 intl. conf. on content-based image and video retrieval. ACM, pp 119–126
24. Yang C, Dong M, Fotouhi F (2005) Semantic feedback for interactive image retrieval. In: MULTIMEDIA'05, pp 415–418
25. Saha SK, Das AK, Chanda B (2007) Image retrieval based on indexing and relevance feedback. Pattern Recogn Lett 28(3):357–366
26. Sobti S, Garg N, Zheng F, Lai J, Shao Y, Zhang C, Ziskind E, Krishnamurthy A, Wang RY Segank: a distributed mobile storage system. In: Proc. of the 3rd USENIX conf. on file and storage technologies
27. Peek D, Flinn J (2006) Ensemblue: integrating distributed storage and consumer electronics. In: OSDI'06, pp 219–232

28. Yan T, Ganesan D, Manmatha R (2008) Distributed image search in camera sensor networks. In: SenSys '08, pp 155–168
29. Gaonkar S, Li J, Choudhury RR, Cox L, Schmidt A (2008) Micro-Blog: sharing and querying content through mobile phones and social participation. In: MobiSys'08, pp 174–186
30. Miluzzo E, Lane ND, Fodor K, Peterson R, Lu H, Musolesi M, Eisenman SB, Zheng X, Campbell AT (2008) Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In: SenSys'08, pp 337–350
31. Pering T, Want R, Gardere L, Vadas K, Welbourne E (2007) Musicology: bringing personal music into shared spaces. In: MobiQuitous, pp 1–8
32. Dearman D, Kellar M, Truong KN (2008) An examination of daily information needs and sharing opportunities. In: Proceedings of the ACM 2008 conference on computer supported cooperative work, pp 679–688