Demo Abstract - MediaScope: Selective On-Demand Media Retrieval from Mobile Devices*

Xing Xu†
University of Southern California
xingx@usc.edu

Yurong Jiang
University of Southern California
yurongji@usc.edu

Peter Terlecky
City University of New York
pterlecky@gc.cuny.edu

Tarek Abdelzaher
University of Illinois at Urbana-Champaign
zaher@illinois.edu

Amotz Bar-Noy
City University of New York
amotz@sci.brooklyn.cuny.edu

Ramesh Govindan
University of Southern California
ramesh@usc.edu

ABSTRACT

Motivated by an availability gap for visual media, where images and videos are uploaded from mobile devices well after they are generated, we explore the *selective, timely retrieval* of media content from a collection of mobile devices. We envision this capability being driven by *similarity-based queries* posed to a cloud search front-end, which in turn dynamically retrieves media objects from mobile devices that best match the respective queries within a given time limit. Building upon a crowd-sensing framework, we have designed and implemented a system called MediaScope that provides this capability. MediaScope is an extensible framework that supports nearest-neighbor and other geometric queries on the feature space (e.g., clusters, spanners), and contains novel retrieval algorithms that attempt to maximize the retrieval of relevant information. From experiments on a prototype, MediaScope is shown to achieve near-optimal query completeness and low to moderate overhead on mobile devices.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.4 [**Information Storage and Retrieval**]: Systems and Software; H.4.0 [**Information Systems Applications**]: General; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms

Design, Experimentation, Performance

Keywords

Crowd-sensing, Image-Retrieval, Feature-Extraction, Mobile-Device

*Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

†The first two authors contributed equally to this work and their names are listed alphabetically by first name.

IPSN'13, April 8–11, 2013, Philadelphia, Pennsylvania, USA.
ACM 978-1-4503-1959-1/13/04.

1. INTRODUCTION

Cameras on mobile devices have given rise to significant *sharing* of media data (photos and videos). Users upload visual media to online social networks like Facebook [1], as well as to dedicated sharing sites like Flickr [2] and Instagram [3]. However, these uploads are often not *immediate*. Camera sensors on mobile devices have been increasing in both image and video resolution far faster than cellular network capacity. Moreover, in response to growing demand and consequent contention for wireless spectrum, cellular data providers have imposed data usage limits, which disincentivize immediate photo uploading and create an *availability gap* (the time between a photo or image is taken and it is uploaded). This availability gap can be on the order of several days.

If media data was available immediately, it might enable scenarios where there is a need for recent (or fresh) information. Consider the following scenario: users at a mall or some other location take pictures and video of some event (e.g. an accident or altercation). An investigative team that wants visual evidence of the event could have searched or browsed images on a photo sharing service such as Flickr to retrieve evidence in a timely fashion.

To bridge this availability gap, and to enable this and other missed opportunities, we consider a novel capability for on-demand retrieval of images from mobile devices. Specifically, we develop a system called MediaScope that permits concurrent geometric queries in feature space on that may be distributed across several mobile devices.

Wireless bandwidth is limited and can vary, *concurrent queries* might compete for limited bandwidth, and query results can be large (since images are large and many images can match a query). These factors can result in unacceptably long query response times, which can impede usability. In some cases, applications might need lower query response times for correctness; in the scenario above, time may be of the essence in taking action (e.g., apprehending suspects).

MediaScope addresses this challenge using an approach that trades off query completeness[1], while meeting timeliness requirements (measured by the time between the issue of the query and when a query result is returned). It incorporates a novel credit-assignment scheme that is used to weight queries as well as differentiate query results by their "importance". A novel credit and timeliness-aware scheduling algorithm that also adapts to wireless bandwidth variability ensures that query completeness is optimized. A second im-

[1]Completeness is intuitively defined as the proportion of desired images uploaded before the timeliness bound.

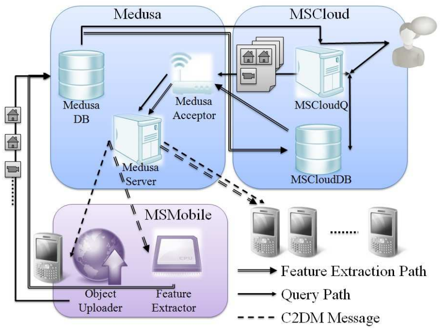


Figure 1—System Architecture Work Flow

portant challenge is to enable accurate yet computationally-feasible feature extraction. MediaScope addresses this challenge by finding sweet spots in the trade-off between accuracy and computational cost, for extracting features from images and frames from videos.

2. MEDIASCOPE DESIGN

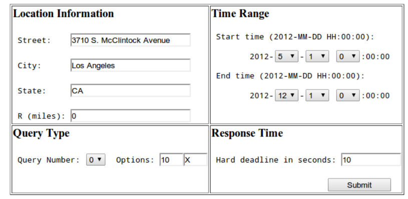
MediaScope is a system that supports timely similarity-based queries on media objects stored on mobile devices. Mediascope is conceptually partitioned across a cloud component called MSCloud, and another component called MSMobile that runs on mobile devices. This partitioned design leverages the computation and storage power of clouds to support geometric queries on the feature space; mobile devices provide sensing and storage for media objects.

These components interact as follows (Figure 1). In the background, whenever participants take photos or videos, the *Feature Extractor* component of MSMobile continuously extracts image and video features and uploads them to the *MSCloudDB*. Users (e.g., a security officer or a reporter) pose queries to MSCloud using a standard web interface, possibly on a mobile device. These queries are processed by the *MSCloudQ* query processing engine, which uses the features stored in the MSCloudDB to compute the query results. The results of the queries identify the media objects that need to be retrieved from individual mobile devices. In some cases, a media object may already have been retrieved as a result of an earlier query; query results are also *cached* in MSCloudDB in order to optimize retrieval. MSCloudQ coordinates with an *Object Uploader* component on MSMobile in order to retrieve query results.

MediaScope uses a publicly available crowd sensing platform called Medusa [8], to enable programmed interaction between MSCloud and Medusa, and to support MediaScope's timeliness requirement, we made several modifications to Medusa Platform. The most challenging component of MediaScope is support for concurrent queries, we designed a *credit assignment mechanism*, the main idea is as follows, each query is awarded a certain amount of credit, and the query itself is responsible for assigned these amount of credit to the qualified objects given by the query optimizer, so each object of uploading request in MSMobile is associated with a credit, MSMobile is going to upload in a way that maximizing the total credit of uploadable objects. Our current instantiation of MediaScope supports three qualitatively different queries: nearest neighbor, clusters, and spanners. Detailed implementation is discussed in the full paper.

3. DEMO DETAILS

We have built MediaScope prototype system using a commodity server machine and Android smartphones. The query interface of MSCloudQ is shown in Figure 2. This demonstration will show the crucial steps of MediaScope: 1) the mobile device once capture



Note:

Query No.	Description	Option 1	Option 2
0	Top K similar	K	Filename @ Server
1	Most Dissimilar (MST)	N/A	N/A
2	Common Interest(K Means)	Num of Clusteres	N/A
3	Cluster Presentation	Num of Clusteres	N/A

Figure 2—Query Interface

an image, the corresponding feature will be extracted and uploaded to MSCloudDB automatically; 2) when MSCloud received query, it will select best media files and ask for uploading; 3) MSMobile will upload media files selected by MSCloud, and then MSCloud return results. For step 2), it is possible that MSCloud get concurrent queries (in the demo, we will issue multiple queries from different tabs of the browser); consequently, MSMobile will receive concurrent uploading tasks, sometimes this means that not all the uploading tasks can be completely uploaded before its timeliness bound and in this situation, the scheduling is critical for the sake of maximizing information (credit) collected.

4. RELATED WORK

There are some other works focus on search over resources, [4] deal with people-centric sensor data; however MediaScope focuses on image search. MediaScope is inspired by leveraging semantics of features [10, 9], techniques for content-based image retrieval from a centralized database of images [11, 6] and image retrieval from mobile devices [7, 5]. Compared to existing works, MediaScope uniquely supports searches on a cloud server, but where the content is stored on the mobile devices and is retrieved on demand.

5. REFERENCES

[1] Facebook. http://www.facebook.com.

[2] Flickr. http://www.flickr.com.

[3] Instagram. http://www.instagram.com.

[4] B. M. Elahi, K. Romer, B. Ostermaie, M. Fahrmair, and W. Kellerer. Sensor ranking: A primitive for efficient content-based sensor search. In *Proc. of ACM Information Processing in Sensor Networks (IPSN)*, pages 217–228. ACM, 2009.

[5] M. Gabbouj, I. Ahmad, M. Amin, and S. Kiranyaz. Content-based image retrieval for connected mobile devices. In *Proc. of Second International Symposium on Communications, Control and Signal Processing (ISCCSP)*. Citeseer, 2006.

[6] C. Jacobs, A. Finkelstein, and D. Salesin. Fast multiresolution image querying. In *Proc. of the 22nd annual conference on Computer graphics and interactive techniques*, pages 277–286. ACM, 1995.

[7] J.S.Hare and P. Lewis. Content-based image retrieval using a mobile device as a novel interface. In *Electronic Imaging 2005*, pages 64–75. International Society for Optics and Photonics, 2005.

[8] M. Ra, B. Liu, T. L. Porta, and R. Govindan. Medusa: A programming framework for crowd-sensing applications. In *Proc. of the 10th international conference on Mobile systems, applications, and services(Mobisys'12)*, pages 337–350. ACM, 2012.

[9] M. Uddin, H. Wang, F. Saremi, G. Qi, T. Abdelzaher, and T. Huang. Photonet: a similarity-aware picture delivery service for situation awareness. In *IEEE 32nd Real-Time Systems Symposium (RTSS'11)*, pages 317–326. IEEE, 2011.

[10] T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In *Proc. of the 8th international conference on Mobile systems, applications, and services(Mobisys'10)*, pages 77–90. ACM, 2010.

[11] K. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *Proc. of the SIGCHI conference on Human factors in computing systems*, pages 401–408. ACM, 2003.