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

Xing Xu[1], Yurong Jiang *[1], Peter Terlecky[2], Tarek Abdelzaher[3], Amotz Bar-Noy[2], and Ramesh Govindan[1]

[1]University of Southern California

[2]City University of New York

[3]University of Illinois at Urbana-Champaign

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 moderate overhead on mobile devices..

1. INTRODUCTION

Cameras on mobile devices have given rise to significant *sharing* of media sensor 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. More important, in response to growing demand and consequent contention for wireless spectrum, cellular data providers have imposed data usage limits, which dis-incentivize immediate photo uploading and create an *availability gap* (the time between when a photo or image is taken and when it is uploaded). To bridge this availability gap, and to enable this and other missed opportunities, we consider a novel capability for selective and timely on-demand retrieval of images from mobile devices. Specifically, we develop a system called MediaScope that permits concurrent timely geometric queries in feature space on media data that may be distributed across several mobile devices. MediaScope is an extensible framework that supports different kinds of queries. MediaScope queries permit nearest-neighbor searches on image feature spaces, *spanners* that retrieve samples of images that span the feature space, or *cluster representatives* that are samples of clusters in the feature space. MediaScope also addresses the timelines and computational complexity issues through the novel concurrent uploading scheme design and accuracy-complexity tradeoff.

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 [10], to enable programmed interaction between MSCloud and Medusa, and to support MediaScope's timeliness requirement, we made several modifications to Medusa Platform. The most challanging component of MediaScope is support for concurrent queries, we designed a *credit assigment 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 first two authors contributed equally to this work and their names appears in the alphabetical order of their first name

IPSN'13, April 8–11, 2013, Philadelphia, Pennsylvania, USA.
Copyright 2013 ACM 978-1-4503-1959-1/13/04 ...$15.00.

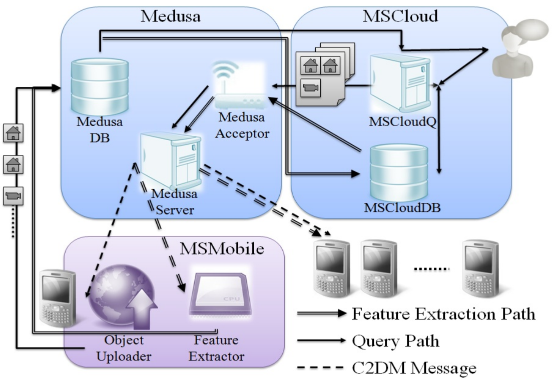
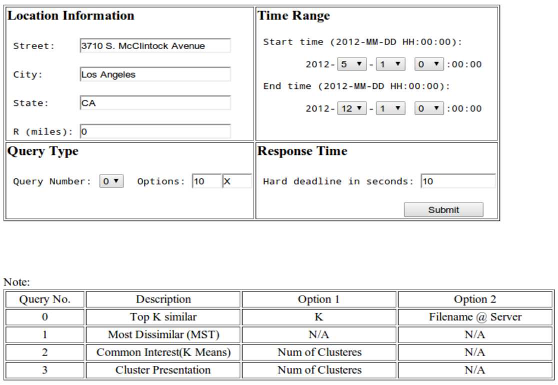


Figure 1—System Architecture Work Flow



Location Information

Street: 3710 S. McClintock Avenue

City: Los Angeles

State: CA

R (miles): 0

Time Range

Start time (2012-MM-DD HH:00:00):

2012- 5 ▾ - 1 ▾ 0 ▾ :00:00

End time (2012-MM-DD HH:00:00):

2012- 12 ▾ - 1 ▾ 0 ▾ :00:00

Query Type

Query Number: 0 ▾ Options: 10 X

Response Time

Hard deadline in seconds: 10

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

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 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, [5] deal with people-centric sensor data; however MediaScope focuses on image search. MediaScope is inspired by leveraging semantics of features [12, 11], techniques for content-based image retrieval from a centralized database of images [13, 6, 8] and image retrieval from mobile devices [9, 7, 4]. 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] I. Ahmad, S. Abdullah, S. Kiranyaz, and M. Gabbouj. Content-based image retrieval on mobile devices. In *Proc. of SPIE*, volume 5684, pages 255–264, 2005.

[5] B. O. M. F. B. M. Elahi, K. R?mer 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.

[6] J. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. Shu. The virage image search engine: An open framework for image management. In *SPIE Storage and Retrieval for Image and Video Databases IV*, pages 76–87, 1996.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.