GlobalSIP 2015

Multimedia Content Creation using Global Network Cameras: The Making of CAM^2

Ahmed S. Kaseb, Youngsol Koh, Everett Berry, Kyle McNulty, Yung-Hsiang Lu, Edward J. Delp
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA
{akaseb, koh0, epberry, mcnulty, yunglu}@purdue.edu, ace@ecn.purdue.edu

***Abstract*—Mobile systems (smartphones and tablets) have become a major channel for consuming and creating multimedia content. Almost half of video-on-demand is watched on mobile systems. Many people use their mobile systems to capture video, and edit, upload, and share content. Using mobile systems to capture video, however, is restricted to users' locations. We believe that the global network cameras could change how multimedia is consumed and created. For example, a traveler may use network cameras to obtain live views of the destination. These cameras can also be rich sources for creating multimedia content—even though the content creator has never visited the locations. This paper presents a system that can retrieve visual data (images or videos) from more than 70,000 network cameras. A user may select cameras and set times for recording the data. It would be possible to make personalized contents around the world without visiting these locations. The paper explains how such a system may provide new opportunities for mobile users and in multimedia creation. The paper also describes some problems that must be solved before the system can be practically deployed.**

I. INTRODUCTION

Mobile systems have profoundly changed how people consume and create multimedia contents. For example, half of YouTube views come from mobile devices [25]. Many people use mobile devices to take videos and share the video clips on social networks. One may join a New Year's celebration, shoot a video, and share the video with friends. Despite the wide availability of mobile devices, some multimedia content is still restricted to professionals only. For example, on a New Year Eve, news agencies send reporters to major cities covering the celebration activities in these cities. Would it be possible for a person to see the celebration of a smaller city (perhaps this person's hometown) that is not covered by any news reporter? The interest to obtain live (i.e., real-time streaming) views (or recent snapshots) extends far beyond New Year Celebrations. For example, photographs play important roles for travellers to select hotels [6]. Including photographs has become common practice in real estate as well. These "static" images, however, have one significant drawback: they are not updated often. In fact, they can be obsolete by several years. Moreover, potential tourists or buyers may want to obtain views at different time, for example, whether a beach hotel has a good view of the sunset. For environment studies, one may wish to observe greenness everyday when studying phenology [20].

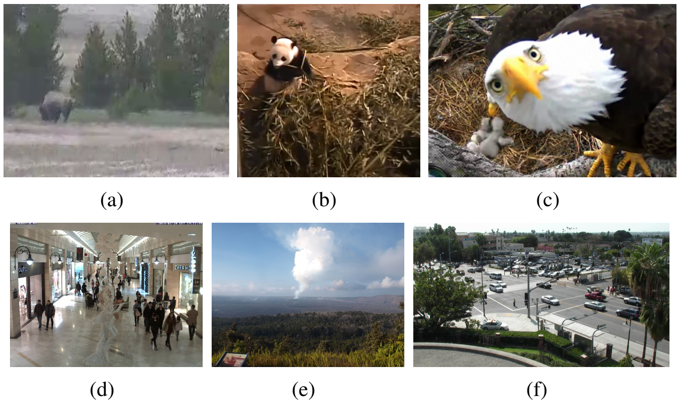
(a) (b) (c)
(d) (e) (f)

Fig. 1. The global network cameras provide opportunities to observe and understand this world. These are some examples. (a) Animal in Yellowstone National Park [22]. (b) Panda Cam [16]. (c) Eagle cam [1]. (d) A shopping mall. (e) Volcano eruption captured by the Hawaiian Observatory on January 10, 2014 [21]. (f) A street in Los Angeles.

These problems may be solved by globally deployed network cameras that continuously stream live views of many places in the world. Many universities, national parks, government agencies, research laboratories, and hotels deploy network cameras and make the data available to anyone connected to the Internet. Figure 1 shows several examples using network cameras to obtain live views of animals' activities, a shopping mall, volcano eruption, and traffic. Live views are important for many reasons, such as

- One may wish to control the time, duration, and frame rate for retrieving the data. Existing street views by map services have only snapshots (no video) but in many situations snapshots are insufficient for creating multimedia content.
- Some snapshots are taken several years earlier and many changes (such as new buildings) may have occurred during these years.
- Some studies require observations over time. One may wish to analyze the demographics of shoppers at different time of a day (morning vs. evening).

II. CAM^2: CONTINUOUS ANALYSIS OF MANY CAMERAS

To explore opportunities for using the data from network cameras, we have been constructing a system called *CAM*^{2} (Continuous Analysis of Many CAMeras, https://cam2.ecn.purdue.edu/). Our previous publications describe different applications and individual components of CAM^2:

- Analyze large volumes of data. One of our experiments retrieve more than 2.7 million images from 1274 cameras in 3 hours at approximately one frame every five seconds or 107Mbps [11]. This experiment used 15 Amazon cloud instances. CAM^2 currently has more than 70,000 cameras. If it retrieves one image from each camera per minute (assuming 100KB per image), CAM^2 can obtain approximately 9TB data per day.

- Provide image-based navigation [13]. We proposed the concept of image-based navigation more than a decade ago [15]. During this decade, image-based navigation has made significant progress: street views have become a popular feature for map services. Existing street views, however, are updated rarely. Densely deployed network cameras could be a potential solution providing frequent updates.
- Conserve energy when mobile systems use image-based navigation [2]. Longer battery life has been one of the most desired improvements for mobile systems. Image-based navigation requires receiving large amounts of data; thus, energy conservation is a crucial factor in realizing the promise of image-based navigation.
- Describe CAM^2's application programming interface [12]. CAM^2 is an open system for researchers. CAM^2 allows users to upload programs analyzing large amounts of visual data. More than 60 users have registered and Purdue University has signed agreements with three universities for sharing Intellectual Properties and source code.
- Teach cloud computing and big data [9], and large-scale image processing [24]. CAM^2 allows students to select data sources to learn image analysis. The students may select sources viewing landmarks or their hometowns.
- CAM^2 uses cloud instances as the computing engines. The large amounts of data require many cloud instances. Some image analysis programs are computationally intense; some other programs are restricted by the memory capacities. CAM^2's resource manager decides the types, numbers, and locations of cloud instances. The resource manager monitors the cloud resources and scales the resources accordingly [14], [4].
- Conduct environmental studies using network cameras [23]. CAM^2 simplifies the process of retrieving data from many sources, as well as resource management for computing engines. The network cameras allow environmental researchers to observe the world [10].
- Different organizations have different rules regarding the amounts of data used to various studies. A paper [18] compares the policies set by the owners of the data.

This paper presents an overview of the entire project, the architecture of the system, the opportunities offered by this system, and the challenges faced in this project, as well as directions for future research using global network cameras for mobile multimedia. The major contributions of this paper include

1) This paper explains the architecture of a CAM^2: how it is designed and how it is built.
2) This paper describes the potential applications of CAM^2: who could benefit using CAM^2 and what types of applications could be built.
3) This paper points out the challenges fulfilling the vision of CAM^2.

A. *CAM^2's Architecture*

Figure 2 illustrates CAM^2's architecture. The major components include

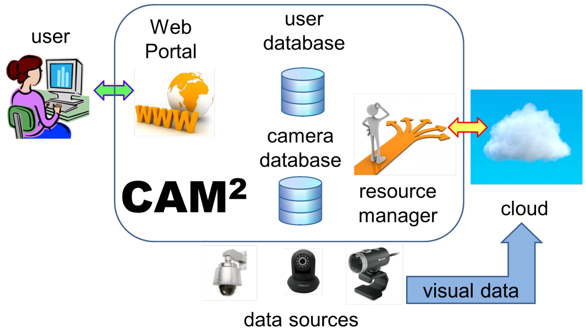

Fig. 2. Architecture of CAM^2. The website is the portal. CAM^2 has two databases for users and for cameras. The resource manager uses cloud for computation and storage. The visual data reach the cloud instances directly, without passing through CAM^2.

1) A camera database stores the the cameras' metadata—their locations, frame rates, protocols to retrieve data, resolutions, owners, etc. Section II-B describes how this database is built.
2) A user database stores their profiles, the configurations, and the uploaded programs.
3) The website is the portal of CAM^2. This portal allows users to select cameras, set configurations, upload programs, and schedule execution. The portal makes the computational engines transparent to the users. As a result, the computational engines can be public cloud or private computers.
4) Cloud instances are used for computation and storage. The data from the cameras go to the cloud directly, without passing through the CAM^2 . This is essential for scalability.
5) A resource manager determines what types of cloud instances to create, where the instances should be located, the duration of the instances, etc. The resource manager plays a crucial role because analyzing large amounts of data needs substantial amounts of resources. The resource manager is also responsible for scaling the resources when needed. The manager migrates tasks to new instances.

CAM^2's API (Application Programming Interface) hides the underlying distributed systems. If an existing program analyzes image (or video) from a file, the needed changes are replacing the statements for input and output in these steps [12]:

1) extending a CAM^2 class called `Analyzer`,
2) replacing the statements for retrieving the data from a file and saving results to a file
3) overriding the event handler to process each new frame

Before starting program execution, a user has to create a configuration that specifies the sources of the data (i.e., the cameras), the desired frame rates, and the duration. This API greatly simplifies the analysis program and the same program can be used to analyze many cameras simultaneously [10].

B. *Camera Discovery*

CAM^2's camera database contains information about more than 70,000 public (i.e., no password protection) network cameras. These cameras can be classified into two categories: (a) The first have known IP addresses and can respond to data requests directly. These cameras are often called "IP cameras" and support HTTP. (b) The second are connected to computers and data requests are handled by these computers. These cameras are often called "webcams".

Finding network cameras can be challenging because there are few repositories that list public network cameras. Searching "network camera" or "webcam" will, unsurprisingly, find the manufacturers of these cameras. Some repositories link to network cameras but these links are often broken. After discovering a network camera, we need to retrieve data from the camera. Different brands of cameras require different methods for retrieving data. Even though IP cameras support HTTP, different paths are needed in the GET commands. Some brands support multiple resolutions while others have only single resolutions. Existing network cameras are designed for human viewers: people would have no problem deciding which button to click for viewing the data but the heterogeneous interfaces make it difficult writing a computer program to retrieve data from different brands. Additionally, some organizations restrict how often data can be retrieved [18]. The policies are written in different formats and currently are processed manually.

Obtaining correct metadata (e.g. latitude and longitude) of network cameras is another challenge [5]. If the metadata of a camera is incorrect, some applications mentioned earlier (e.g. image-based navigation) would not work correctly. Finding accurate metadata of network cameras can be difficult. Even though metadata is provided by some camera owners, the information is sometimes incorrect. Geo-coding services can convert street addresses to geo-locations [7, 8] but many camera owners do not provide street addresses. Moreover, using IP addresses to look up geographical locations is not as reliable as desired [17]. One major contribution of CAM^2 is the significant effort creating the camera database (some of locations are found manually). To our knowledge, this is the largest database of geo-tagged network cameras.

C. Resource Management

CAM^2 allows users to analyze live data or to store the data for offline analysis. One main advantage of using CAM^2 is the built-in resource management [14], [4] using cloud instances when analyzing data. Cloud computing allows users to rent instances by hours. Each cloud vendor offers multiple types of instances with different numbers of cores, amounts of memory, geographical locations, price per hour. Selecting the most cost-efficient instances can be a complex optimization problem because many factors must be considered. Moreover, different computer vision programs have different characteristics. Some programs are limited by computation and others are limited by memory. For the former, selecting a cloud instance with more cores would achieve noticeable performance improvement but a cloud instance with more memory would have negligible effects. For the latter, the opposite is true. If high frame rates are desired, a cloud instance with a short distance (measured by the network round trip time) to a camera should be selected [11]. To complicate the matters, the prices of cloud instances at different locations may differ by 40% even though they have the same number of core and the same amount of memory [3].

Some cloud vendors provide autoscaling and load balancing—launching new cloud instances when the running instances are over-utilized and terminating running instances when the the utilization is low. After scaling the instances, load should be re-balanced among the instances. However, existing load balancing by cloud vendors is inapplicable for a system like CAM^2. Load balancing means distributing the workload to running cloud instances. Existing load balancing is designed for web requests with short durations. Moreover, existing solutions balance loads by redirecting new incoming requests (e.g. HTTP, HTTPS, SSL) to different instances. For CAM^2, an analysis program may last hours or even longer and the incoming data are images or videos, not HTTP, HTTPS, or SSL. As a result, existing solutions cannot balance the load and CAM^2 must handle load balancing based on the resource utilization and performance requirements. Long-running programs cannot be easily migrated if these programs analyze live streaming data, e.g. counting the number of vehicles on a road. Migration could cause disruption (failing to count vehicles) or duplication (counting the same vehicles twice) and must be handled carefully. CAM^2's current resource management monitors the utilization of all cloud instances used for analyzing the data. If an instance is over-utilized, another instance is launched and some data streams are moved to the new instance. If two or more instance are under-utilized, the data streams are consolidated and the excessive instances are terminated. This is a "best-effort" approach in resource management: disruption and duplication may occur during migration. We are investigating how to prevent disruption and eliminate duplication.

III. FUTURE APPLICATIONS OF CAM^2

Since the first public release in July 2014, CAM^2 has demonstrated its ability in many applications mentioned above. We envision that many more applications can take advantage of CAM^2. This section describes several examples. We classify these examples into three groups based on the required densities of network cameras.

The first group requires only a few network cameras—as few as only one—to be useful. Such applications are likely to be scientific research. Cameras are deployed for specific purposes, such as observing wildlife, monitoring air quality, or volcanos at specific locations. Even though the data are public and everyone can see the data, we hypothesize that the data from these few cameras are useful to only a small number of researchers. The second group requires more cameras, perhaps hundreds inside the downtown of a city. Traffic cameras deployed by departments of transportation can fulfill this purpose. Figure 3 shows the view of a parade from two traffic cameras in New York City. By using dozens of traffic cameras, one may see the parade from multiple angles. Another application is showing live views of vacation resorts to


(a) (b)

Fig. 3. A parade seen from two different traffic cameras.

attract tourists. As network cameras become fairly inexpensive (an HD network camera costs less than $100USD), we expect more hotels to provide live views. In the near future, providing live views could be as common as providing snapshots now.

The third group requires densely populated network cameras. These cameras will enable new applications, such as real-time virtual tours and image-based navigation. It is likely that these cameras are owned by different organizations and adopt different protocols. Finding a uniform way to communicate with these heterogeneous cameras would be a non-trivial task. Classifying these cameras based on content (e.g. indoor vs. outdoor or city vs. nature) would be another challenge [19]. It is conceivable that many new applications could be invented if such a system exists. The impacts to mobile multimedia could be profound. To name a few, mobile users would be able to obtain live view of many places all around the world. Many people already use their mobile phones to select restaurants when they travel. If live views are available, these users can know, in real-time, whether the restaurants are crowded. The global network cameras can also become rich sources for creating multimedia content. Consider the example shown in Figure 3, one may create a report of the parade without visiting New York.

IV. CONCLUSION

The global network cameras provide many opportunities for creating and consuming multimedia content. Before realizing this vision, we must solve some difficult problems, including discovering network cameras, obtaining their metadata, correctly and efficiently managing resources, and offering an abstraction hiding the details so that it is easy for users to use the system. This paper presents CAM^2—a system for retrieving, visualizing, and analyzing visual data from global network cameras. We give an overview of the architecture of the system, including a web portal for users to upload analysis programs, a database of over 70,000 cameras, and a resource manager for dividing analysis workloads among cloud instances for scalable computing. Readers are encouraged to visit https://cam2.ecn.purdue.edu/ and register as users. For collaboration, please contact Yung-Hsiang Lu (yunglu@purdue.edu).

ACKNOWLEDGMENTS

The authors would like to thank the organizations that provide the data. A complete list of the data sources is available at https://cam2.ecn.purdue.edu/acknowledgements. CAM^2 uses Amazon Elastic Cloud and Microsoft Azure as the computing engines. This project is supported in part by National Science Foundation ACI-1535108, IIP-1530914, and CNS-0958487. Koh is supported by the Lynn CSE Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] Decorah Eagle Cam Live Feed. http://www.decoraheaglecamalerts.com/.

[2] S. M. I. Alam, S. Fahmy, and Y.-H. Lu. LiTMaS: Live road Traffic Maps for Smartphones. In *IEEE Workshop on Video Everywhere*, 2015.

[3] Amazon. AWS Pricing. http://aws.amazon.com/pricing/.

[4] W. Chen, Y.-H. Lu, and T. J. Hacker. Adaptive cloud resource allocation for analysing many video streams. In *IEEE International Conference on Cloud Computing Technology and Science*, 2015.

[5] J. Choe, T. Pramoun, T. Amornraksa, Y.-H. Lu, and J. D. E. Image-Based Geographical Location Estimation Using Web Cameras. In *Southwest Symposium on Image Analysis and Interpretation*, pages 73–76, 2014.

[6] A. Dickinger and J. Mazanec. Consumers' preferred criteria for hotel online booking. In P. OConnor, W. Hpken, and U. Gretzel, editors, *Information and Communication Technologies in Tourism 2008*, pages 244–254. Springer Vienna, 2008.

[7] FFIEC. Ffiec geocoding system. https://geomap.ffiec.gov/FFIECGeocMap/GeocodeMap1.aspx.

[8] Google. Google geocoding api. https://developers.google.com/maps/documentation/geocoding/.

[9] T. J. Hacker and Y.-H. Lu. An instructional cloud-based testbed for image and video analytics. In *Emerging Issues in Cloud Workshop of CloudCom*, 2014.

[10] N. Hemsoth. On the api for harvesting global camera networks. The Platform, http://www.theplatform.net/2015/04/02/an-api-for-harvesting-global-camera-networks/, April 2 2015.

[11] A. S. Kaseb, E. Berry, Y. Koh, A. Mohan, W. Chen, H. Li, Y.-H. Lu, and E. J. Delp. A system for large-scale analysis of distributed cameras. In *IEEE Global Conference on Signal and Information Processing*, 2014.

[12] A. S. Kaseb, E. Berry, E. Rozolis, K. McNulty, S. Bontrager, Y. Koh, Y.-H. Lu, and E. J. Delp. An interactive web-based system for large-scale analysis of distributed cameras. In *Imaging and Multimedia Analytics in a Web and Mobile World*, 2015.

[13] A. S. Kaseb, W. Chen, G. Gingade, and Y.-H. Lu. Worldview and route planning using live public cameras. In *Imaging and Multimedia Analytics in a Web and Mobile World*, 2015.

[14] A. S. Kaseb, A. Mohan, and Y.-H. Lu. Cloud resource management for image and video analysis of big data from network cameras. In *International Conference on Cloud Computing and Big Data*, 2015.

[15] Y.-H. Lu and E. J. Delp. An Overview of Problems in Image-Based Location Awareness and Navigation. In *Visual Communications and Image Processing*, pages 102–109, 2004.

[16] S. N. Z. Park. The Giant Panda Cam. http://nationalzoo.si.edu/animals/webcams/giant-panda.cfm.

[17] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye. IP Geolocation Databases: Unreliable? *SIGCOMM Comput. Commun. Rev.*, 41(2):53–56, Apr. 2011.

[18] L. C. Pouchard, M. S. Nelson, and Y.-H. Lu. Comparing policies for open data from publicly accessible international sources. In *Annual Conference International Association for Social Science Information Services & Technology*, 2015.

[19] T. Pramoun, J. Choe, H. Li, Q. Chen, humrongrat Amornraksa, Y.-H. Lu, and E. J. D. III. Webcam classification using simple features. In *Computational Imaging*, 2015.

[20] A. Richardson, J. Jenkins, B. Braswell, D. Hollinger, S. Ollinger, and M.-L. Smith. Use of digital webcam images to track spring green-up in a deciduous broadleaf forest. *Oecologia*, 2007.

[21] N. P. Service. Hawai'i Volcanoes National Park. http://www.nature.nps.gov/air/webcams/parks/havocam/havocam.cfm.

[22] N. P. Service. Old faithful geyser streaming webcam. http://www.nps.gov/features/yell/webcam/oldFaithfulStreaming.html.

[23] W.-T. Su, Y.-H. Lu, and A. S. Kaseb. Harvest the information from multimedia big data in global camera networks. In *IEEE International Conference on Multimedia Big Data*, 2015.

[24] W.-T. Su, K. McNulty, and Y.-H. Lu. Teaching large-scale image processing over worldwide network cameras. In *IEEE International Conference on Digital Signal Processing*, 2015.

[25] Youtube. Youtube statistics. https://www.youtube.com/yt/press/statistics.html.