Vision: Mapping the World in 3D through First-Person Vision Devices with Mercator

Pieter Simoens
Ghent University College
Valentyn Vaerwyckweg 1
B-9000 Gent, Belgium
psimoens@intec.ugent.be

Tim Verbelen
Ghent University - iMinds
G. Crommenlaan 8 bus 201
B-9050 Gent, Belgium
tverbelen@intec.ugent.be

Bart Dhoedt
Ghent University - iMinds
G. Crommenlaan 8 bus 201
B-9050 Gent, Belgium
dhoedt@intec.ugent.be

ABSTRACT

Many vision-based applications, especially in the domain of augmented reality, must align the camera position in the observed scene. However, traditional cameras only register textures of the observed scene. The reconstruction of depth information from 2D images is compute intensive and inevitably results in loss of accuracy. In this article, we present Mercator, a cloudlet-based system to build a 3D model of the world on which other applications can be built. The model is continuously updated, refined and expanded by crowd-sourcing depth data from 3D cameras on head-mounted devices such as Google Glass. Mercator scales up to a worldwide system by distributing the model over the network edge in geographically close cloudlets. We present the software building blocks of the system, and discuss challenges related to data management, privacy and programming model.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding; C.2.4 [**Distributed Systems**]: Distributed Applications

Keywords

crowd-sourcing, wearable computing, 3D map, cloudlet, computer vision, smartphone

1. INTRODUCTION

Of all sensors embedded in today's smartphones, the camera provides the richest form of information. Despite the omnipresence of cameras in smartphones, the variety in vision-based applications has been extremely limited for reasons of usability and computational requirements. Whereas sensors like gyroscopes or GPS provide their data without noticeable effort of the user, cameras must be manually held in the right position. Moreover, today's mobile devices lack the computational power to fully tap into the vast amount of information that is embedded in visual sensor input.

The emergence of elegant and light head-mounted devices (HMD), such as Google Glass, and the convergence between mobile and cloud computing may however enable a new wave of vision-based applications [6]. The form factor of HMDs allows for continuous, effortless capture of first-person viewpoint video, while compute intensive algorithms can be offloaded with low latency owing to the advances in mobile broadband and the emergence of cloud infrastructure at the network edge.

Most vision-based applications require a 3D structural model of the observed scene and spent many processing cycles to infer this model from the 2D texture information that is captured by commodity cameras. Twodimensial computer vision can only derive *relative* measures of depth in the scene, but no *absolute* distances between objects. The reconstruction of depth information inevitably results in loss of accuracy as it is intrinsically limited. Objects can be detected through contour identification, but without depth information it is hard to ascertain whether the user is looking at the actual object or a picture of that object.

In this paper, we present our vision on Mercator, a distributed system to create a 3D map of the complete world by crowd-sourcing depth information streamed from sensors mounted on HMDs. Figure 1 illustrates our vision. Each movement of a user, no matter how small, results in depth data from a different perspective that makes the map more accurate, complete and up-to-date. Setting aside computational and memory constraints, Mercator will eventually result in a high-resolution, up-to-date 3D map of the complete world.

For reasons of latency and scalability, we distribute the 3D map over different cloudlets. Cloudlets are distributed cloud infrastructure at the network edge and have been proposed for offloading real-time applications from mobile devices [20]. Beyond this original motivation, we exploit data locality and store on each cloudlet only the data of its immediate geographical environment. Real-time applications of users that have been offloaded to the same cloudlet consequently have fast, reliable access to the data of nearby scenes.

New hardware and software technologies have emerged that lower the barriers to realizing our vision. Both the cost and size of depth cameras have dropped to allow their integration into HMDs. PrimeSense [5] unveiled earlier this year the Capri sensor, a System on a Chip that is 10x smaller than today's depth sensors, small enough to fit in today's smartphones. An alternative are stereoscopic cameras such as Go-

MCS'13, June 25, 2013, Taipei, Taiwan

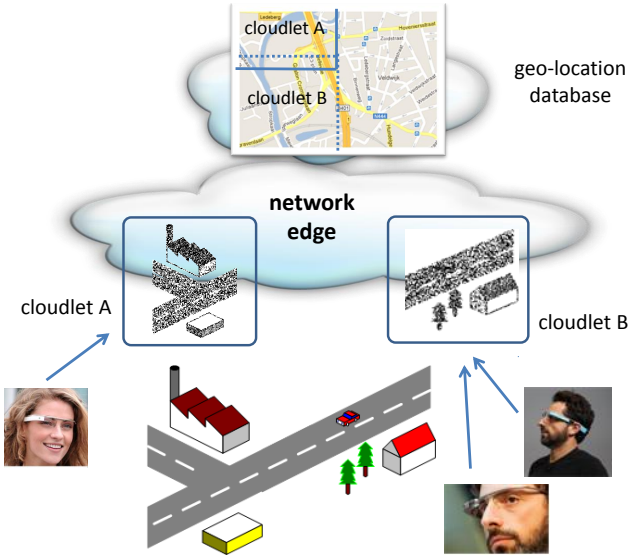

Figure 1: Depth sensors on head-mounted devices continuously update and refine the 3D model of the world. The model is distributed over multiple cloudlets, according to geographical proximity.

Pro 3D [2]. A recent milestone in the domain of computer vision was the demonstration of Kinect Fusion [19], a GPU-based software framework reconstructing in real-time geometrically precise 3D models of medium-sized indoor scenes based on the depth data of handheld Kinect sensors.

In the rest of this paper, we expose our vision to taking these new technologies to a global scale. In section 2, we speculate on innovative applications enabled by the Mercator framework. In section 3, we elaborate on key aspects of the Mercator architecture. The major challenges related to data retrieval, user privacy, programming model and spatial and temporal data accuracy are discussed in section 4.

2. APPLICATIONS

The 3D map of the world provides a cloud-based equivalent of the observed reality. In this section, we list a few applications in the domain of augmented reality that exploit Mercator for accurate placement of virtual objects or for annotating physical objects with metadata.

Cognitive assistance: where are my keys? Users can carry a virtual purse in their smartphone with 3D models of personal objects. These models can be used to locate lost objects. Instead of asking everybody if they have seen your keys, you simply upload the 3D representation of your keys for matching against the 3D map. People who have quickly glanced over your keys are probably unaware that they actually have seen your keys, but their gaze has been registered in the cloud and provides the necessary information to guide you to your lost keys.

Personal tagging We envision Mercator to provide functionality to augment the 3D model with one or more multiple virtual layers. Users attach virtual objects or tags to specific parts in the model. Systems like Airbrush [17] have already demonstrated virtual tagging of real-life objects, but the work is limited to a single user. With Mercator, multiple users can query the 3D model for virtual objects added by others that are within their current field of view.. Imagine leaving a virtual scribble on the fridge, automatically rendered on the HMD of each person entering the kitchen. Each object becomes a potential pervasive display for communicating with others [16].

Pervasive object manipulation Apart from adding virtual objects, we can also expand the 3D model with metadata on physical objects, such as their interface, or their IP address. Applications query the model to locate available sensors and actuators in the immediate environment of the user and to retrieve the interface to manipulate these. By aligning the current gaze to the Mercator data set, a vision-based application can retrieve the metadata of the object the user is currently looking at. Until now, pervasive interaction with physical objects has mainly been restricted to confined, well-controlled environments, such as Google's Interactive Space [1], where the IP address and interface of objects is known. Through Mercator, any authorized user can enter a room and communicate with any networked object, by retrieving the IP address and the appropriate interface from Mercator's metadata.

Indoor navigation Despite many years of research, no single solution has yet crystallized that is as elegant, accurate and simple as GPS for outdoor environments. Many approaches to indoor positioning use signal readings, such as Wi-Fi [18] fingerprints or FM signals [8]. These signals are however unstable over time and not intrinsic to the structure of the indoor environment. Vision-based indoor localization closely resembles the human orientation mechanism and is based on relative positioning against the structural landmarks of the scene. Good results with depth cameras have been demonstrated in the field of robotics [7].

3. ARCHITECTURE

The current cloud infrastructure model is based on a limited number of large datacenters. Services can rapidly scale up by cloning VMs inside the datacenter. This model is not suited for the type of the vision-based services presented in the previous section, which are personalized and depend on real-time processing of high-bandwidth streams with very low-latency to large numbers of geographically distributed users. In our architecture, presented in Figure 2, applications running inside personal VMs access the distributed model data through tailored Mercator components.

3.1 Cloudlets for low-latency

Increasing the coverage of users with low-latency access to cloud infrastructure requires a substantial increase of data centers that must be deployed closer to the user. To illustrate the problem, in [9] researches have experimentally verified that Amazon EC2 can only serve 70 % of the US population with a latency low enough for cloud-based gaming [9]. To overcome network latency, the VM-based cloudlet model has been widely studied [20]. Note that cloudlets were originally positioned as soft-state mini-clouds, but we assume that Mercator components, e.g. to merge incoming depth data with the model, are persistently running on the cloudlet. This requires the cloudlet infrastructure to be managed by network operators who may charge the users of Mercator, either as part of their monthly subscription plan or implicitly via advertising. This vision is becoming a reality: IBM and Nokia Siemens Networks recently introduced CloudServer: service hosting infrastructure fully integrated

with the base station of mobile networks [3]. Note that the base station is the network location closest to the user that is still under control of the network operator. Revenues can be generated through user licensing, or by selling advertisements on virtual billboards in the Mercator data set.

3.2 Data locality

The regions of interest to a user are heavily correlated with his current geographical location, either to upload new scene perspectives or to import the model data of the immediate surroundings in their own vision-based applications. Scenarios with users exploring more remote regions, such as map navigation, are typically less stringent on latency.

This data locality can be exploited by spatially distributing the model data over cloudlets. This strategy ensure the tractability of the 3D computer vision algorithms for aligning new sensor input and *fusing* this data into the model, or for monitoring changes to the map by tracking individual objects. Such computer vision algorithms typically scale linearly with the spatial size and accuracy of the scene model. As an example, Microsoft Kinfu realizes real-time mapping of 3D data of a *single* Kinect by keeping the complete model in GPU memory, trading off memory efficiency for speed. The surface of the earth not covered by water is about 150 million km^2. Assuming an average mapping height of 10 m (more in cities, less in rural areas) with a medium resolution of 4 bytes per voxel of 1 cm^3, the complete model of the world requires approx. 6 zettabytes of memory. Hence, Mercator's scalability can only be provided by distributing the data model, especially since the GPU memory on the cloudlet is not only needed for aligning uploaded depth data with the model, but also for the vision-based applications running inside the personal VMs.

3.3 Public and private data

Spatially distributing the model data does however not address the privacy requirements of users. Some parts of the world should not be fused into the global model accessible to everyone. Many of the speculative applications described in section 2 require sharing of personal, typically indoor environment models with a strictly defined subset of users. As shown in Figure 2, we distinguish between public and private scenes.

Mercator must provide the necessary authorization and authentication mechanisms to regulate access to different parts of the data set. Turning Mercator into a successfull service requires users trusting the system to preserve the integrity of their private models. Given the popularity of cloud-based services like Dropbox for storing personal files, we advocate that most users have no strong objections against storing data in the cloud *an sich*. However, the biggest challenge of trust is to prevent users from unintentionally sharing sensitive data with other users, without hampering the usability of the system. In section 4, we elaborate on this trade-off between privacy and user friendliness.

3.4 Personal Virtual Machines

Each user instantiates a personal VM on the cloudlet, running computationally heavy and latency sensitive applications. We assume this personal VM contains the Mercator library, with specific components to interface with the 3D model. On the one hand, users must install privacy filters inside their personal VM, that forwards sensor data to the

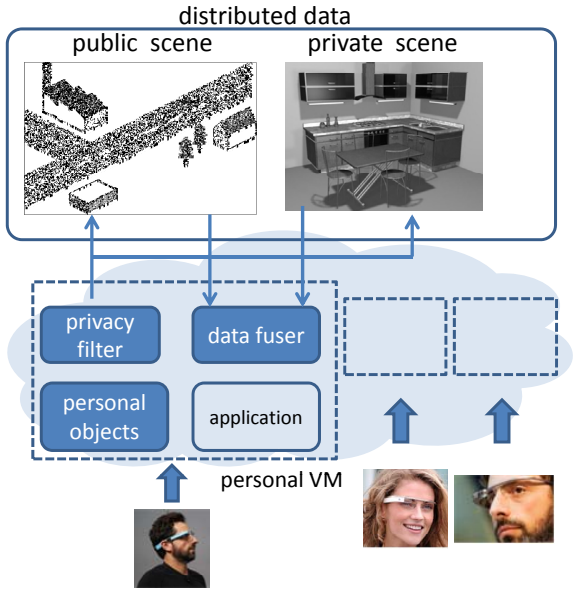

Figure 2: Cloudlet-based architecture of Mercator.

public or private parts of the model. On the other hand, the *data fuser* must hide the underlying complexity of distributed access to the application.

The data fuser synthesizes the requested scenes from both public and private 3D models. Note that as users move to other places, data will be needed stored on other cloudlets. As the latency of data retrieval increases, the data fuser might initiate a migration to another cloudlet. Live migration of personal VMs is a challenging issue. However, through a series of optimization techniques, the time for provisioning such a customized VM from a baseline image has been reduced to approx. 10 s [11].

Mercator also offers a software component to manage the models of personal belongings (such as your keys). These models never leave the personal VM, turning this VM into an electronic purse. The models of your personal belonging are automatically carried with you as you move.

4. CHALLENGES

The computer vision domain has evolved significantly in the past decade. Tracking the camera position in a twodimensional map was first demonstrated in 2007 [15], and in 2011 real-time tracking in a 3D map in medium-sized rooms up to 7 m^2 was realized by KinFu [13].

Given these impressive advancements in the domain of computer vision, we will focus on the challenges in other domains that must be overcome to extend Kinfu-alike single-user, indoor systems to the scale of Mercator: a multi-user, trustable global 3D map service that can be used indoor and outdoor. In the remainder of this section, we discuss challenges related to capturing depth data, preserving privacy, ensuring model accuracy and the need for a good programming model.

4.1 Map accuracy

Point clouds are the narrow waist of the data models used in 3D computer vision algorithms. For reasons of scalability and speed, higher-level surface models are inferred from point clouds, such as meshes or surfels [12]. To support the broadest range of applications, Mercator must (at least)

provide a detailed point cloud of the complete world. The accuracy of this point cloud model involves both spatial (resolution) and temporal (lifetime) aspects.

The ultimate goal of Mercator is to provide each possible perspective on the world. Unfortunately, no depth cameras exist with a depth of field matching the human gaze, which may stretch several kilometers. Mercator must support multiple input formats, including depth maps from Kinect-like sensors and 3D videos. Depth measurement techniques like time-of-flight or infrared scanning do not suffer from adverse lightning effects, but the random measurement error increases quadratically with the distance from the sensor [14]. In outdoor environments, other sensor techniques will be needed to crowd-source the map with high-resolution data. One solution is to infer depth information from stereoscopic 3D cameras such as the GoPro 3D [2]. The concept may build on existing work such as Photosynth [4] that creates a navigable 3D map by stitching multiple pictures. However, such an approach requires powerful post-processing and accuracy can be hurted in low-light conditions, through reflections, etc.

The world is continuously evolving, with objects being replaced and people moving around. By fusing new perspectives into the model each time users glance over a scene, the map is kept temporally consistent. However, providing up-to-date maps for a busy street with fast moving cars is extremely challenging. The paradox is that the more devices are observing a scene, the more accurate the map will be. On the other hand, the movement of these users will cause more dynamism and result in more changes. We therefore propose to maintain short-term and more stable models. Mercator must gradually learn to distinguish between structural elements with a fixed position in the world, and objects that are continuously being replaced. For example, if an object is placed on the table, Mercator should retain the data about the covered area for fast surface reconstruction once the object is removed.

4.2 Energy overhead

Continuously sensing and uploading depth data inevitably stresses battery lifetime of the wearable device. The Kinect sensor generates a depth map with a resolution of 640x480, a framerate of 30 fps and a depth of 16 bpp, resulting in a raw data stream of 150 Mbps. Although depth data is captured in a pixel grid similar to RGB color information, 2D video codecs cannot directly be applied because of the random noise introduced by depth measurement techniques such as time-of-flight or infrared scanning. State-of-the-art depth compression techniques apply adaptive depth coding, allocating more bits to active regions, to reduce the bitstream to approx. 3 Mbps [10]. Assuming the energy cost of data upload to 1 J/MB [21] and the Samsung Galaxy S4's battery capacity of 2600 mAh, the upload of depth data would drain the battery in less than 7 hours. The actual battery autonomy will even be shorter owing to the power needed for continuous sensing and compression.

An additional complexity is that lowering the fidelty of the uploaded depth data directly impacts the accuracy of the computer vision algorithms. Only when lossless compression is applied, the depth data will be accurate enough to be fused into the global model. Apart from lossless compression, adding application-layer intelligence can mitigate the power consumption. For example, we could correlate the density of the uploaded depth map to the lifetime of the observed scene in the cloudlet model. If a scene has just been refreshed, there is no need to upload depth data with full fidelity. If a device runs low on energy, the density of the uploaded depth maps can be temporarily reduced to the lowest level needed for tracking the camera position in the observed scene. An optimal trade-off must be found between reducing the load for individual users and the global goal of keeping the 3D map up to date.

4.3 Virtual curtains and fences: privacy

We discern two major questions regarding privacy. Mercator must provide strict guarantees about access rights to the models of private environments. More fundamentally, the question raises about how to prevent people unintentionally uploading data through their HMD.

People passing by your house should not have access to a detailed model of its inside. As stated earlier, we conjecture users to trust their cloudlet infrastructure provider that the model of sensitive spaces is not publicly exposed. The major challenge is rather to design a non-intrusive user interface for configuring access rights to private scenes. Applications must not constantly bother users with authorization requests popping up in their HMD, yet users want to keep track about who uses the model of their property. In a first version of Mercator, we will apply surface detection on the model to detect when users enter a new room and query the user if no access rights have been configured.

As no one can prohibit people to take images at public places, it is nearly impossible to avoid that people upload models of public or semi-public places. Therefore, we would propose an opt-out model for Mercator, similar to Google Streetview where house owners can ask to blur the model of their facade. Conversely, models of private domains are only included upon explicit request of the owner.

The biggest privacy thread stems from the fact that others might, perhaps unconciously, upload models of your interior through their always-on HMD. In the past, recording was an explicit act and required to take the smartphone out of your pocket. Social control was mostly sufficient to refrain people from recording private conversations. Wearable head-mounted devices have a radically different user interface that makes recording nearly effortless and unconscious: it only involves pressing a button on the shank of the glasses. Chances are that people will not notice when they are being recorded through the HMD of their correspondent. We believe that strong legislative regulation will be needed, possibly inspired by the existing legislation on portrait rights.

4.4 Programming Model

Mercator must provide appropriate interfaces for read and write operations on the data set. This includes a powerful interface to specify access rights on subsets of the data. To ensure scalability, Mercator should intelligently provide only relevant parts of the data set and associated metadata when serving read requests from applications. Possibly, data must be fused from multiple VMs and cloudlets. All other functionality, such as aligning the user's viewpoint with the Mercator data set for camera tracking is assumed to be application specific.

The global model will be divided in submodels with their own access rights. Each submodel contains multiple pointers to the location of adjacent models. Applications requesting data from Mercator must prove their identity through

a certificate issued by a central registration authority. In a pilot trial, we would aim to infer submodels automatically through surface detection. Like the physical world, private property is demarcated through walls, fences and curtains. This should prevent occassional passers-by in the street from looking behind the facades. Users must then manually configure access rights for each room. Based on these experiences, we might evolve to semi-automatic algorithms for configuring user access. For example, users might configure Mercator to grant access to everybody entering their house.

Apart from providing an API, Mercator will design software blocks to annotate the 3D model with metadata. The same levels of authorization apply: some objects may be shared with the public, whereas others may only be shared with relatives. A distinction should be made between read and write operations. Allowing everybody to add metadata to public parts of the world will immediately clutter the map. Service providers can restrict write rights to specific places and/or to (paying) instances. In private places, social control might suffice to keep the map clean.

5. CONCLUSIONS

The emergence of head-mounted devices with an elegant form factor allows for nearly effortless and continuous capture of the carrier's first person viewpoint. In this paper, we have explained our vision towards Mercator, a system for building a 3D map of the world by crowd-sourcing data from depth cameras mounted on HMDs. Many compelling applications can be built on top of this system, from indoor navigation to advanced augmented reality. The global map is distributed over different cloudlets and split in public and private parts. In addition to structural information, the map can contain metadata about physical and virtual objects.

Many open questions remain, especially pertaining to privacy, which we believe can only be tackled through a combination of pilot trials, user surveys and legislative actions. By building Mercator, we hope to bring valuable contributions in this domain. The challenge to preserve privacy is however not specific to Mercator, but fundamentally originates from the fact that people now have the ability to continously and almost unnoticeable record their interactions with other people. We invite researchers to think along this way.

6. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable feedback. Tim Verbelen is funded by a Ph.D. grant of the Fund for Scientific Research in Flanders (FWO-Vlaanderen).

7. REFERENCES

[1] Google Interactive Spaces. An API and Runtime for creating interactive, physical spaces, 2013. `https://code.google.com/p/interactive-spaces`.

[2] Gopro 3d hero system, 2013. `http://gopro.com/hd-hero-accessories/3d-hero-system`.

[3] IBM and Nokia Siemens Networks Announce World's First Mobile Edge Computing Platform, 2013. `http://www-03.ibm.com/press/us/en/pressrelease/40490.wss`.

[4] Microsoft Photosynth, 2013. `http://www.photosynth.net`.

[5] PrimeSense shows off tiny Capri sensor, yearns for 3D-sensing future (hands-on), 2013. `http://www.engadget.com/2013/01/11/primesense-capri-hands-on`.

[6] P. Bahl, M. Philipose, and L. Zhong. Vision: cloud-powered sight for all: showing the cloud what you see. In *Proc. of the third ACM workshop on Mobile cloud computing and services*, 2012.

[7] J. Biswas and M. Veloso. Depth camera based indoor mobile robot localization and navigation. In *Robotics and Automation, IEEE International Conference on*, 2012.

[8] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. Fm-based indoor localization. In *Proc. of the 10th international conference on Mobile systems, applications, and services*, MobiSys, 2012.

[9] S. Choy et al. The brewing storm in cloud gaming: A measurement study on cloud to end-user latency. In *Network and Systems Support for Games, 2012 11th Annual Workshop on*, pages 1–6. IEEE, 2012.

[10] J. Fu, D. Miao, W. Yu, S. Wang, Y. Lu, and S. Li. Kinect-like depth data compression. *Multimedia, IEEE Transactions on*, 2013.

[11] K. Ha, P. Pillai, W. Richter, Y. Abe, and M. Satyanarayanan. Gigasight: Scalable crowd-sourcing of video from mobile devices. In *Proc. of the 11th international conference on Mobile systems, applications, and services*, MobiSys, 2013.

[12] P. Henry et al. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *the 12th International Symposium on Experimental Robotics*, volume 20, pages 22–25, 2010.

[13] S. Izadi et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proc. of the 24th annual ACM symposium on User interface software and technology*, 2011.

[14] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[15] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Japan, 2007.

[16] T. Kubitza, S. Clinch, N. Davies, and M. Langheinrich. Using mobile devices to personalize pervasive displays. *Demo. at HotMobile*, 12, 2012.

[17] M. Marner, B. Thomas, and C. Sandor. Physical-virtual tools for spatial augmented reality user interfaces. In *IEEE International Symposium on Mixed and Augmented Reality*, pages 205–206, 2009.

[18] W. Meng et al. Secure and robust wi-fi fingerprinting indoor localization. In *Indoor Positioning and Indoor Navigation (IPIN), International Conference on*, 2011.

[19] R. A. Newcombe et al. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE International Symmposium on Mixed and Augmented Reality*, 2011.

[20] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. *Pervasive Computing*, 8(4):14–23, 2009.

[21] P. Simoens et al. Gigasight: Scalable crowd-sourcing of video from mobile devices. In *Proc. of the 11th international conference on Mobile systems, applications, and services*, MobiSys, 2013.