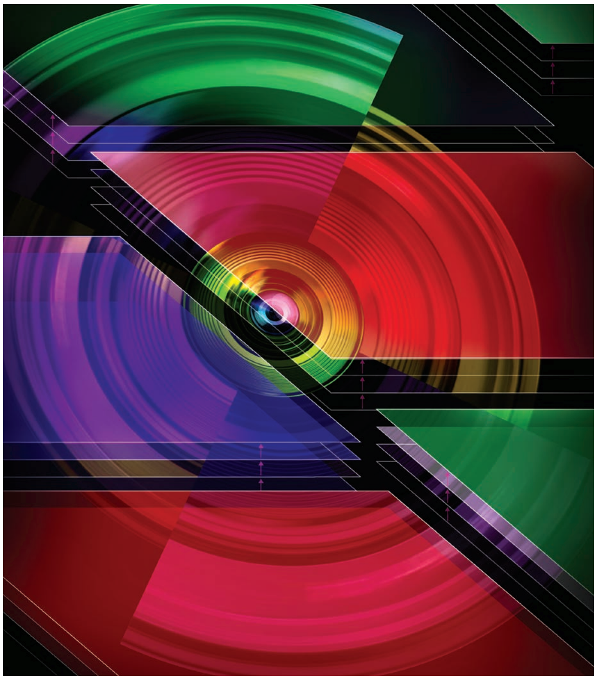

Integrating Consumer Smart Cameras into Camera Networks: Opportunities and Obstacles

Andrea Prati, *University IUAV of Venice*
Faisal Z. Qureshi, *University of Ontario Institute of Technology*

Integrating personal camera-equipped mobile devices in traditional camera networks could enable a new class of mobile vision applications, but although integration is becoming technically feasible, consumer buy-in could be a thorny problem.

Camera networks, which comprise passive and pan-tilt-zoom devices, have received more attention in recent years,[1] in large part because of their suitability in a range of applications, from surveillance to urban planning. This increased focus has come in concert with an explosion of smart camera devices, including smartphones, tablets, and most recently Google Glass. Not surprisingly, the number of mobile vision algorithms to capture, analyze, and store images has also grown.

However, current mobile vision applications treat each camera device in isolation, ignoring the potential of collaborative sensing, in which multiple devices work together in a network designed to optimize energy consumption and image quality. To date, this emphasis is understandable: smart cameras continue to evolve, and researchers have only recently begun to address their role as a sensor node in an existing network infrastructure. But newer mobile devices are certainly equipped to handle sensing tasks. Adding these devices to a traditional camera network or video surveillance infrastructure could usher in a whole new class of applications, from crowd-sourced and people-centric surveillance to participatory event recording.

To answer technical questions about the possibility of integration, we looked at research in mobile vision, network calibration, time synchronization, energy-efficient algorithms, and the dynamic use of mobile devices and identified a range of technical challenges in each area that suggest directions for additional research. Perhaps the most critical issue to resolve is the form of consumer participation: Will mobile device users be willing to join the network, for how long, and given what incentive? Given the dynamic nature of these devices, it is natural to question if a camera network can ever rely on consumer devices for its sensing needs. Although the network cannot rely solely on consumer devices, it can definitely use the images that they capture, and often to great advantage, as in disaster recovery.

To our knowledge, we are the first to explore in depth the issues in integrating camera-equipped consumer devices into ad hoc networks that already comprise a mix of

Published by the IEEE Computer Society

passive and active cameras. Our aim is to produce a compendium of research opportunities and open problems in this area.

MOBILE VISION

Mobile vision is the development of computer vision and image-processing techniques for camera-equipped consumer devices.[2] Mobile vision shares and builds on work in active vision, which deals with algorithms for processing the images that active (nonstationary) cameras capture. Cameras in handheld devices are also active, but their motion is harder to model than the motion from an active camera on a robot, which in part has earned handheld camera-equipped devices the separate category of mobile vision. Vision algorithms that deal with egomotion, or visual odometry—the process of determining a robot's position, orientation, and velocity by analyzing associated camera images—are relevant to both active and mobile vision.

Mobile vision also poses unique challenges that stem from mobile device limitations in processing, memory, and bandwidth. Power requirements are of particular concern in implementing vision algorithms on consumer devices. Cameras mounted on small/light unmanned vehicles (UAVs) share many characteristics with cameras embedded in consumer devices.

Ultimately, by their sheer ubiquity, consumer devices will shape mobile vision techniques that meet the needs of highly mobile, transient smart cameras with restricted computational and energy resources. UAV cameras will benefit, of course, but they will not drive the change.

Device requirements and new protocols

Camera-equipped consumer devices now have the computational resources to support computer vision algorithms, along with sensors such as compasses, accelerometers, and gyroscopes that can support applications such as augmented reality and geolocalization.

The last 30-plus years of research on computer vision algorithms have produced highly accurate and efficient algorithms that consumer devices are sophisticated enough to handle. Even if the algorithms exceed the device's computational capacity, it can use high-speed, low-latency wireless communication protocols to draw on cloud-based computational resources to reduce that burden.

Applications and frameworks

Software development tools, standard computer vision libraries, such as OpenCV (www.opencv.com) or Qualcomm FastCV (https://developer.qualcomm.com/mobile-development/mobile-technologies/computer-vision-fastcv) and graphics libraries, such as OpenGL have made it easier to develop mobile applications for consumer devices. Open source operating systems, such as Android, and augmented reality platforms, such as VRToolKit, have also facilitated development.

Consequently, myriad mobile vision projects and products are appearing in object recognition (landmarks, logos, and goods), augmented reality, gesture recognition, road sign recognition, lane detection, automatic cruise control, and collision avoidance.[2] The applications and frameworks in the "What's New in Mobile Vision?" sidebar reflect some interesting developments.

Some of these applications and frameworks follow the client-server model, in which a cloud server relieves the camera device of the more complex image computations. The degree to which computation is split between device and server depends on the algorithm and its complexity, dataset size, available bandwidth, and the application's performance requirements.

INTEGRATION ISSUES

None of the new mobile vision offerings consider the possibility of multiple consumer devices cooperating in common sensing or imaging tasks, nor do they acknowledge the possibility of integrating consumer devices into existing surveillance networks. Some researchers have proposed schemes in which mobile phones are stationary nodes in a network of pan-tilt-zoom cameras,[3] but such proposals fail to recognize that mobility is precisely why integration is challenging. For instance, unlike PTZ cameras, the movement of mobile phones cannot be controlled in a prespecified way to support a sensing goal.

The core issue for integration is that, to carry out collaborative sensing, each node must be able to cooperate with others, including any mobile devices. In a typical application scenario, a camera network, such as that in Figure 1, must identify events of interest in a particular area, which requires answering where and when questions. To answer the where question, nodes in the camera network must have a common coordinate system (geometric calibration). To answer the when question, they must be able to synchronize clocks (time synchronization).

Thus, adding mobile devices to a camera network will require algorithmic optimization, as well as addressing significant geometric calibration and time synchronization issues. Table 1 lists key mobile device features and the pluses and minuses for integration.

Integration also requires looking at energy efficiency, the dynamic nature of mobile device use, and other challenges, such as security and privacy and scalability.

Algorithmic optimization

Mobile devices require sensing algorithms that are opportunistic, leveraging mobile phones in the scene when it is possible to do so. Reengineered algorithms are not likely to be sufficient support in this regard. Rather, a new class of vision algorithms is required to analyze

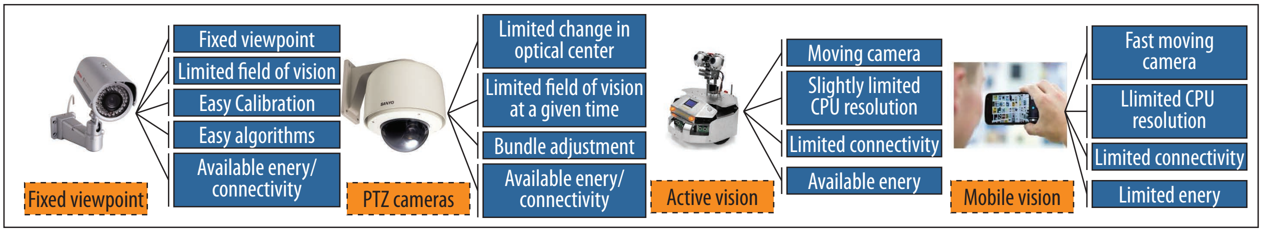

Figure 1. Characteristics of cameras that could be part of an integrated camera network. The challenge is to address the differing needs and requirements for these devices, particularly between pan-tilt-zoom (PTZ) cameras and consumer mobile devices.

Table 1. How mobile device features and requirements could influence an integrated camera network.

Feature or requirement	Pluses	Minuses
Time synchronization with other nodes	Many clock sources available	Delays from communication over unreliable connections could affect synchronization strategy
Spatial synchronization with other nodes	Additional embedded sensors (such as gyroscope and GPS) can help calibration	Difficult to make any assumption or have static calibration or common coordinate system
All-in-one devices (computation, communication, sensors, display)	No need to assemble with communication (Wi-Fi/Zigbee) modules, or displays	Hard to add additional modules to improve the system
Freely moving camera with all the degrees of movement	—	Challenge for computer vision algorithms
Limited power	—	Battery capability crucial for user satisfaction; limits applications
Ability to capture specific people-centric view of a scene	New applications; perspective that can help in surveillance, disaster recovery, and so on	Privacy concerns (difficult to convince users to donate their mobile phone resources)

the images that transient, highly mobile consumer devices capture. Any such algorithm must account for mobile devices' characteristics that affect image capture and analysis.

Real-time performance versus limited resources. Typical mobile vision applications are interactive by design and hence require real-time performance. However, most mobile computing platforms have limited computational resources. Enabling real-time performance in a resource-constrained environment might require designing algorithms from the ground up. One solution is to optimize data access, particularly to data stored in caches, by preallocating data, avoiding data duplication, and exploiting temporal and spatial data locality. Another approach is to use low-level code optimization, such as loop unrolling and pipelining, and to use hardware-specific instruction sets.

Memory is also limited, and processing can be slow. Data types such as floating-point calculations are not compatible with the devices' lack of dedicated or specialized hardware, and certain algorithms perform poorly in limited formats. Thus optimization should include smaller data types and limit processing to small data subsets.

Motion and energy use. Mobiles devices exhibit rapid, jittery motion, which makes it difficult for any algorithm to analyze the captured images. Even if the algorithm could deal with these images, it is likely to have a large energy footprint.

Communication. A key feature of mobile devices is their ability to communicate with each other and with a central server. Client-server algorithms that divide the work between the device and a server are a promising way to offload computational needs. Again, however, energy consumption from increased communication overhead is a concern.

Geometric calibration

In some camera network applications, the nodes must establish a common coordinate system. Video surveillance, for example, requires multiple observations of the same person or object from different cameras. In these applications, a common coordinate system enables the object's partial 3D reconstruction and greatly simplifies object matching across multiple views. However, the device's extrinsic parameters—position and orientation—must be calibrated, which often comes with high computational and energy requirements.

Proposed algorithms. Calibration algorithms to estimate a mobile device's extrinsic parameters do not exist in part because it is not clear how to constrain the device's 3D location. Standard chessboard-based approaches are suitable only for estimating the intrinsic parameters of cameras on mobile devices. Partial calibration techniques that estimate planar homography rely on finding correspondences in the scene, but they assume that the camera is processing images continuously in real time to update the correspondences needed for those estimates.

One decentralized calibration approach[4] assumes that the camera constantly processes the captured images to extract features, which are used in a decentralized and collaborative way to calibrate the network. The approach also assumes that camera views overlap, which is not always the case.

Another approach[5] uses extremely low resolution images to calibrate the network. A location sensor, such as an ultrasound Cricket receiver, encodes the locations of four reference points (correspondences). Four low-resolution pictures of a reference point, each taken from a different view, are sufficient to estimate the location and the orientation of a visual sensor node. The geometric calibration process uses these estimates to coordinate sensor nodes.

The method assumes that sensor nodes are stationary and that calibration is performed only when the network is set up, but by deploying multiple reference points in the scene, it might be possible to use the method to estimate a mobile device's extrinsic parameters. However, there are still concerns. For example, a large number of reference points must be present in the scene, and reference points must be visible at all times. This isn't always possible because of occlusions. The calibration's quality and accuracy will depend on both the number of reference points visible in an image and the image resolution.

An alternate approach for camera network calibration is to exploit the other sensors embedded in mobile devices, such as gyroscopes, GPS, compasses, and accelerometers. Location sensors, such as GPS or Wi-Fi-based localization, can provide a coarse estimate of a device's location. Recent work even uses encoded LED lights to estimate the location of mobile devices.[5]

Although these approaches can only roughly estimate a device's location and orientation, refinements are possible with computer vision techniques. Unfortunately, energy consumption is still an open problem. Mobile devices have limited energy budgets, so cameras on mobile devices are unsuitable for continuous use.

Requirements for new algorithms. Maintaining the calibration of a camera network with highly mobile nodes is computationally expensive. One direction is to develop algorithms that can perform multicamera sensing without relying on camera and network calibration. Another is to search for new methods and techniques that can estimate high-quality camera and network calibration, since some applications, such as scene 3D reconstruction or structure-from-motion will always require complete 3D camera network calibration.

Time synchronization

In collaborative sensing, a common clock allows the fusing of information captured at multiple nodes, yielding a more complete picture of the event in question. Even without fine-grained time synchronization, it is often desirable to time-order lower-level events to exploit causal relationships and detect higher-level events.

Sensor network research has addressed time synchronization in ad hoc wireless networks in some depth. One important finding[6] is that classical clock synchronization algorithms are unsuitable for wireless sensor networks because each node has a limited communication range and all the nodes are highly mobile. Both reasons make continuous clock sharing infeasible.

Classical time synchronization algorithms rely on estimating communication delays between different nodes, which is difficult when the nodes are highly mobile. The timestamp transformation scheme[6] circumvents this problem by using local clocks to generate timestamps, which nodes share. The receiving node transforms the timestamp to its local time. The timestamp scheme assumes that

- each node has a clock with a known maximum clock drift, and
- the two nodes sharing timestamps remain connected long enough to exchange one additional synchronization message.

Although this method works well for ad hoc wireless sensor networks, it might not be straightforward to deploy on camera networks with mobile devices because the second assumption would be hard to ascertain. Even so, adapting the timestamp scheme is worth investigating, possibly by having the centralized server perform timestamp transformation, similar to the way a cell tower handles time synchronization in mobile devices.

Learning-based synchronization is another option. Such schemes attempt to model communication delays statistically by relying on previous observations. However, communication delays depend heavily on the camera network's current status, which cannot be known a priori. For this option to work in general, research must address different time synchronization needs through application-specific time synchronization schemes.

Energy efficiency

Applications that run on camera-equipped mobile devices must be energy efficient. Although limiting data acquisition is one way to achieve that efficiency, it is not ideal because both performance and interactivity benefit from constant data acquisition and processing. A mobile vision algorithm must balance sporadic and continuous data acquisition, and mobile vision research has explored several principles to achieve that balance. These principles should guide any methods, techniques, and tools for developing energy-aware computer vision algorithms[7–9]:

- *Limit active time and increase idle time.* This principle applies to both processing steps and sensor activation. In hierarchical processing, the mobile device CPU can be activated only when new data come in or at the user's input, while remaining idle for the rest of the time. In hierarchical sensor activation, sensors with small energy requirements are on all the time and can activate sensors with higher energy requirements only when needed.
- *Use hardware-specific code optimization.* The algorithm can turn off hardware units when they are not needed, saving energy and prolonging the device's operational life.
- *Use the display efficiently.* Using the screen on a mobile device drains energy. Algorithms should not reflect on the display intermediate results—rather, only final and significant ones, to keep it switched off for more time.
- *Exploit multicore architectures.* Many mobile devices have multicore processors that can handle single-instruction, multiple-data parallelism. Moreover, exploiting data parallelism is intrinsic to computer vision and image processing algorithms.

Dynamic nature of mobile devices

Unlike other camera nodes, a mobile device is not completely subservient to the network; rather, it is there to serve its user. The user can remove the mobile device from the camera network without any warning; the mobile device is not always on or pointed in the right direction; and the user can be employing the device in a way that is unrelated to the sensing or processing that the camera network requires.

We see two ways to work around this erratic availability. This first is to have the mobile consumer device initiate the connection to the camera network and offer its imagery and storage and processing resources. Suppose, for example, that an individual sees an event of interest and starts recording this event on his mobile device. The device can query other cameras, both stationary and nearby mobile devices, and begin a collaborative sensing session.

The second way is to have the camera network initiate contact with mobile devices in the vicinity of an event of interest. This scenario is somewhat more problematic because the camera network must convince the user to employ his device in this way.

Both options require new network management and control theory and methods, as well as new techniques for supporting collaborative sensing using a multitude of imaging sensors each with different imaging, sensing, processing, and communication capabilities.

The dynamic availability of mobile devices must be resolved before integration is possible. Users will most likely have to give consent to network participation, and they will need an incentive to capture images for the camera network. Other needs are new techniques for opportunistic sensing that will enable a camera network to use mobile devices as they appear in the scene and redesigned scheduling and resource allocation algorithms that account for the network's high dynamicity. Obviously, much work remains in this area.

Other integration challenges

Some integration issues do not fit neatly into the categories we have described so far. These include network topology estimation, security and privacy, and network size and scale.

Network topology. Because mobile devices exhibit unpredictable and extreme motions, estimating network topology is often difficult. Standard approaches[1] include centralized processing, in which every node communicates with a central server; distributed processing, in which nodes share information with other nodes nearby; and clustered processing, in which nodes are in clusters and clusters share information.

Centralized processing is perhaps the best starting point for estimating topology in a camera network with mobile devices. Another option might be to set up spatially oriented clusters of mobile devices, formed according to device proximity estimates from GPS or Wi-Fi localization. Setting up and maintaining these clusters could introduce communication lag.

A fully connected network might have prohibitively high bandwidth and energy requirements. However, topology estimates are one way to more efficiently use communication resources and reduce energy use.

Security and privacy. Smart camera networks and ubiquitous video surveillance raise serious social, ethical, privacy, and legal concerns, and integrating camera-enabled consumer devices into these networks multiplies those concerns.[10,11] The aim is to protect both the privacy of an active participant in the camera network and the privacy of those the participant could exploit. Fortunately, the same computer vision algorithms that make video surveillance so intrusive are also useful in implementing privacy protection. This is a promising direction for future research.

Integrating consumer mobile devices into video surveillance networks opens an entirely different way to interact with the cameras on the scene. Current video surveillance networks are closed systems that the few use to observe the many. There is no way for an observed individual to know how the video is collected, who has access to it, and what happens to it.

Proponents of video surveillance claim that closed networks ensure individual safety. We do not dispute this claim, but having access to a sanitized video from a surveillance system can also be a benefit. Consider a mother who loses a child in a busy mall. In such extreme circumstances, shouldn't the parent have immediate access to the video feeds through her mobile phone?

Privacy and legal issues do come into play, but this scenario is an example of how camera-equipped mobile devices can enable new ways for consumers to interact with and benefit from an existing camera network. Such applications are fertile ground for user interaction and security research.

Network size and scale. Because thousands of camera-equipped consumer devices might become nodes at any given time, an integrated network could become quite large. Camera networks not only cover extended spaces but also must contend with density from a small space with many mobile device users. Such variance and scale will require new tools to study issues in extremely large networks.

What's New in Mobile Vision?

These applications and frameworks give a flavor of the directions that mobile vision applications are taking:

- CarSafe uses rear and rear-facing front cameras for in-vehicle applications. The rear camera monitors distances from other vehicles and tracks lane changes. The rear-facing front camera tracks the driver's head position and blink rate as indicators of microsleep, drowsiness, and distraction (http://now.dartmouth.edu/2012/09/dartmouth-smartphone-app-targets-driver-safety/).
- iOnRoad warns the driver of frontal collision or lane departure. It uses Qualcomm's FastCV mobile-optimized computer vision library. It also monitors headway and can identify and locate other cars in the driver's field of view (www.ionroad.com).
- Leafsnap is an electronic field guide that uses visual recognition software to help identify tree species from photographs of their leaves (www.leafsnap.com).
- Nokia Point and Find (http://betalabs.nokia.com/trials/nokia-point-and-find) and Google Goggles (www.google.com/mobile/goggles/) link image-based queries to a Web search. Google Goggles, for example, can use a query image to find matching images and text in Google's image database.
- Qualcomm Vuforia is a generic software development kit for augmented reality and object recognition. (www.qualcomm.com/solutions/augmented-reality).
- Snaptell is a visual recognition application for books, DVDs, and games (http://snaptell.typepad.com/).
- Word Lens (http://questvisual.com/us/) and Google Translate for mobile (www.google.com/mobile/translate/) are applications for automatic text recognition.

Recent developments in processer technologies and embedded systems are equipping consumer devices with enough processing to become nodes in a network of active and passive cameras. We believe that it is both feasible and desirable to proceed with this kind of integration. However, the sheer scale and density of these networks, combined with the dynamicity of the consumer devices and their extreme, unrestricted motion, are beyond existing techniques to set up ad hoc smart camera networks.

We have identified a host of interesting technical challenges: low-power, energy-efficient computer vision processing; camera network calibration and time synchronization; camera network control, coordination and scheduling; security and privacy; and user interaction. Our current focus is on low-power, energy-efficient computer vision processing on mobile devices. This ability, we feel, underpins some of the other areas of research identified in this article.

Ours is one perspective on integration, but we expect to see others explore the idea of using networks for participatory sensing, video surveillance, and urban monitoring. Opportunities abound for both fundamental and applied research to answer technical, ethical, and social questions, taking the next important steps toward moving ad hoc smart camera networks from vision to reality.

References

1. H. Aghajan and A. Cavallaro, eds., *Multicamera Networks: Principles and Applications*, Academic Press, 2009.
2. G. Hua et al., "Introduction to the Special Issue on Mobile Vision," *Int'l J. Computer Vision*, vol. 96, no. 3, 2012, pp. 277–279.
3. P. Bolliger, M. Köhler, and K. Römer, "Facet: Towards a Smart Camera Network of Mobile Phones," *Proc. 1st Int'l Conf. Autonomic Computing and Communications Systems*, 2007, p. 17:1–17:10.
4. D. Devarajan, Z. Cheng, and R. J. Radke, "Calibrating Distributed Camera Networks," *Proc. IEEE*, vol. 96, no. 10, 2008, pp. 1625–1639.
5. X. Liu et al., "Snapshot: A Self-Calibration Protocol for Camera Sensor Networks," *Proc. 3rd IEEE Int'l Conf.*

Broadband Communications, Networks and Systems, 2006, pp. 1–10.
6. K. Römer, "Time Synchronization in Ad Hoc Networks," *Proc. 2nd ACM Int'l Symp. Mobile Ad Hoc Networking & Computing*, 2001, pp. 173–182.
7. B. Dieber, L. Esterle, and B. Rinner, "Distributed Resource-Aware Task Assignment for Complex Monitoring Scenarios in Visual Sensor Networks," *Proc. 6th Int'l Conf. Distributed Smart Cameras* (ICDSC 12), 2012, pp. 1–6.
8. B. Dieber, C. Micheloni, and B. Rinner. "Resource-Aware Coverage and Task Assignment in Visual Sensor Networks," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 21, no. 10, 2011, pp. 1424–1437.
9. B. Rinner et al., "Resource-Aware Configuration in Smart Camera Networks," *Proc. Computer Vision and Pattern Recognition Workshop* (CVPRW 12), 2012, pp. 58–65.
10. J. Fan et al., "A Novel Approach for Privacy-Preserving Video Sharing," *Proc. 14th ACM Int'l Conf. Information and Knowledge Management* (CIKM 05), 2005, pp. 609–616.
11. M. Kirkup and M. Carrigan, "Video Surveillance Research in Retailing: Ethical Issues," *Int'l J. Retail Distribution Management*, vol. 28, no. 11, 2000, pp. 470–480.

Andrea Prati *is an associate professor in the Department of Design and Planning in Complex Environments at the University IUAV of Venice. His research interests include computer vision and multimedia, image analysis and processing, and distributed smart cameras. Prati received a PhD in computer engineering from the University of Modena and Reggio Emilia. He is a senior member of IEEE and a member of ACM and GIRPR (Italian branch of the International Association of Pattern Recognition). Contact him at andrea.prati@iuav.it.*

Faisal Z. Qureshi *is an associate professor of computer science and the founding director of the Visual Computing Laboratory at the University of Ontario Institute of Technology, Canada. His research interests include sensor networks, computer vision, and computer graphics. Qureshi received a PhD in computer science from the University of Toronto. He is a member of IEEE, ACM, and Canadian Image Processing and Pattern Recognition Society. Contact him at faisal.qureshi@uoit.ca.*