

# Cloud-Assisted Smart Camera Networks for Energy-Efficient 3D Video Streaming

Zhangyu Guan and Tommaso Melodia,  
*State University of New York at Buffalo*

**Despite current obstacles, smart camera networks that offload computationally intensive tasks to remote servers could allow energy-efficient 3D and multiview video encoding and delivery yet still ensure high-quality multiple-device video streaming.**

**E**merging 3D, multiview, and stereoscopic video services such as 3D cinema or free-viewpoint video can offer a considerably higher quality of experience than conventional 2D video. Smart camera technologies will soon make possible similarly novel services for mobile users, including 3D video capture and display, multiview wireless surveillance, and even glimpses of a 3D ocean through an underwater acoustic network.

The tradeoff for innovations like these is computational intensity. More elaborate videos can consist of up to several hundred 2D views,<sup>1</sup> requiring encoding and transmission that quickly drain a smart camera's battery. Any solutions to this energy problem must stem from equally innovative transmission schemes and energy-efficient network architectures that support high-quality 3D wireless video streaming in smart camera networks.

## SMART CAMERA NETWORKS

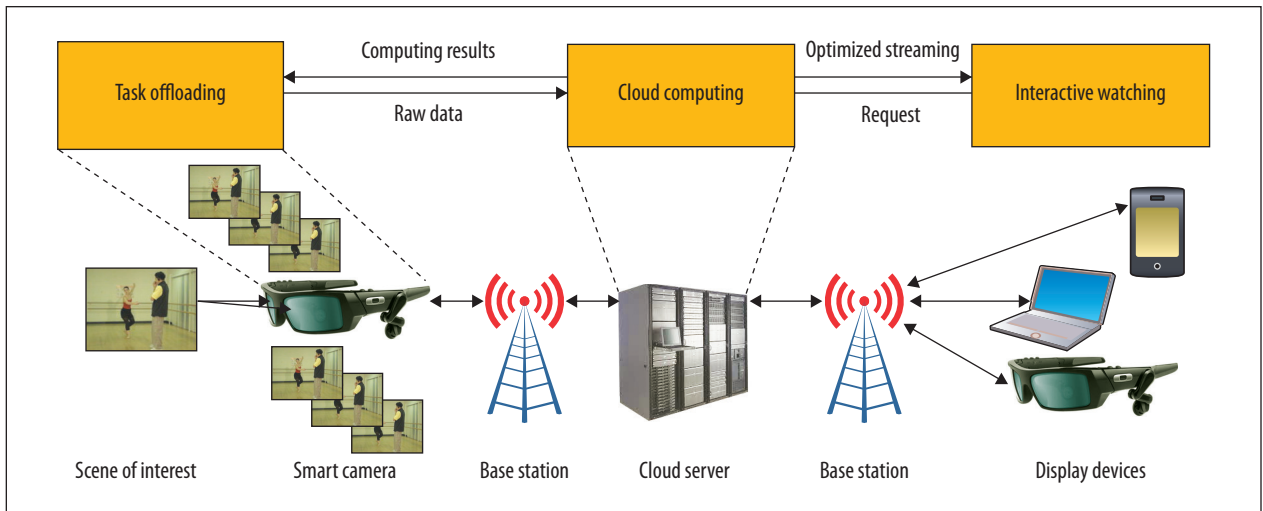
Figure 1 shows one such vision of an architecture that integrates smart cameras, video streaming,

and multimedia cloud computing. Cameras continuously offload computationally intensive tasks to a remote cloud server, potentially extending battery life. The server computes the data and returns results without compromising compression efficiency or video quality.

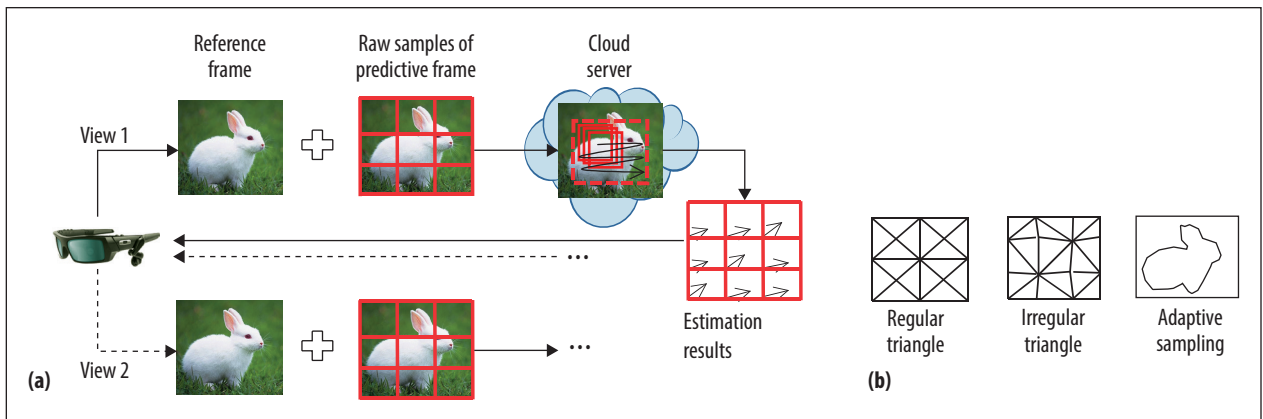
However, to accommodate a network of multiple cameras and display devices, the cloud server must do more than process data. Users watching 3D videos are apt to interactively request video content in a specific view. The cloud server must consider all these views concurrently as well as differences among display devices to achieve optimal high-quality scalable video streaming in real time.

To better understand the challenges for developing such architectures and possible solutions that meet these requirements, we reviewed current work in cloud-enabled smart camera networking and 3D wireless video streaming. We looked at architectural components that support cloud-assisted video encoding on the client side, cloud-based video decoding on the server side, and scalable cloud-client networking.

Most of the approaches we found are based on predictive encoding, which is promising but requires complex computations and suffers from choppy video streaming in wireless environments. To address these drawbacks, we are exploring video networking based on compressive sampling, a clean-slate approach to video capture, encoding, and decoding in a smart camera network.



**Figure 1.** Envisioned architecture to support 3D video streaming in the cloud. The cloud server not only computes raw data but also optimizes streaming across individual cameras and display devices.



**Figure 2.** Simulcast video encoding. The camera offloads encoded computation to the cloud server by (a) transmitting only the reference frame in its entirety and raw samples of the ensuing predictive frames according to (b) a regular or irregular triangular mesh that adapts to the object in the image. The server returns computation results to the camera, which then finishes the encoding.

## VIDEO ENCODING

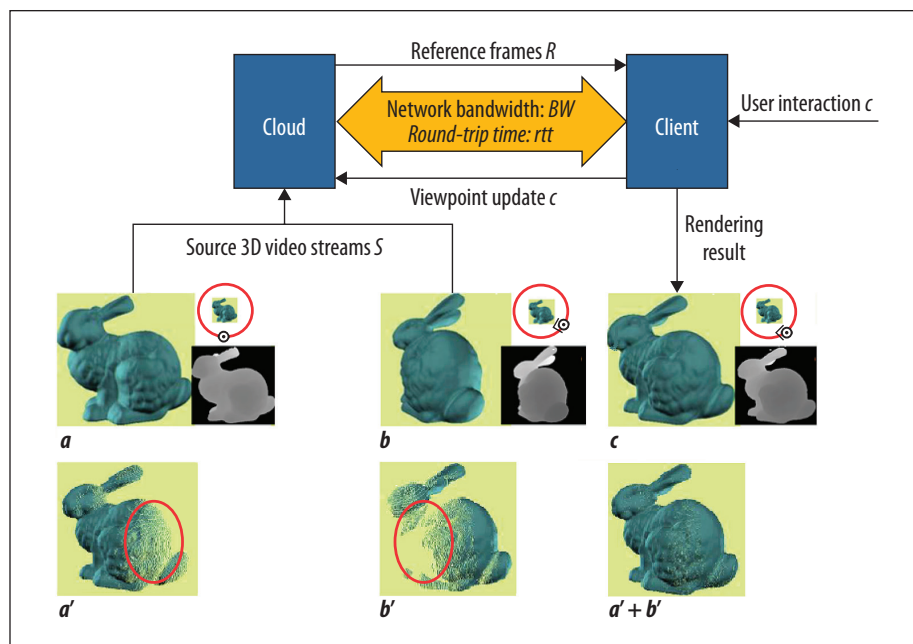
To exploit bandwidth efficiently, video delivery systems encode raw sequences using signal processing techniques to remove spatial and temporal redundancy. Encoding 2D video is already complex; motion estimation operations, for example, can account for 90 percent of total encoding complexity.

Given that 3D, multiview, and stereoscopic videos consist of at least two 2D views (for stereoscopic recording) and may run to hundreds of views (for free-viewpoint applications),<sup>1,2</sup> encoding complexity can easily exceed most current mobile camera resources. Sony's Vaio VGN-UX1XN mobile device offers a case in point: working at full power using traditional 2D encoders just to encode different views independently, the Sony could process only two to five stereo video frames

per second (fps)—far from the 30 fps needed for real-time video.<sup>1</sup> Complexity increases even more when the system must encode correlated views at the same time.

## Complexity offloading

Consider the video encoding in a single smart camera. To enable 3D and multiview encoding, a smart camera can offload computationally rich data to the cloud server in two ways. It can either send feature information, such as the point cloud of the scene's surface, which the server then uses to reconstruct the original scene, or it can send a small portion of raw video data, which the server then uses to run predefined complex computations, returning the computation results to the camera as the basis of further encoding the scene.



**Figure 3. Architecture for cloud-assisted rendering.** The cloud server considers two reference views, and rather than transmit a color pixel frame for each view, it generates a depth map (gray frames). The camera uses the map to place each pixel in its coordinate system for that view, thereby obviating the need to download additional raw data. Red ellipses depict occlusion artifacts, which the fusion of the two frames overcomes.

Figure 2 depicts the independent encoding and transmission of a two-view simulcast video, where a traditional monoview video encoding format such as MPEG-4, H.264, or their high efficiency video coding (HEVC) successor is used to encode each view in parallel. For example, a group of pictures representing each view might consist of a reference frame followed by 30 predictively encoded frames. Encoding involves complex motion estimation between this reference frame and each predictive frame. Offloading requires the camera first to encode the reference frame independently and transmit it to the cloud server, which reconstructs it as a reference for motion estimation. Then the camera uploads only a sample of each ensuing predictive frame. Figure 2b shows sampling that follows a regular or irregular triangular mesh.<sup>3</sup> (An irregular mesh is based on a regular mesh, but is more adaptive to image objects.)

The cloud server estimates motion (distance and direction) by comparing these uploaded raw pixels with the reference frame to search for the best match. Finally, it returns estimation results to the camera for actual video encoding.

Experiments in a wireless local area network (WLAN) using triangular mesh sampling<sup>3</sup> proved that, relative to local motion estimation (in the camera), offloading consumes approximately 30 percent less energy. More sophisticated sampling methods, such as sampling the boundary of moving objects, could further decrease the camera's energy use.

## Challenges and possible solutions

Other techniques have proven viable in reducing video encoding costs. For example, preliminary studies show that computing the disparity among views locally reduces video encoding by half.<sup>1</sup> However, computing view disparities in the cloud creates problems because the server has only part of the original predictive frame. These down-sampled frames often make view correlation impossible because the predictive frame of one view cannot serve as the reference for other views. Uploading additional raw data to preserve correlation negates any energy savings and so is not a solution.

An alternative allows cameras to translate motion estimation results for one

view to their own coordinate systems, using geometric information such as the camera's relative position and the scene's depth to correlate views without uploading additional raw data. More experiments with this alternative are required to determine its feasibility.

Cloud computing for 3D video encoding operates under the same limitations as any multimedia cloud computing network. A task is not worth offloading if doing so consumes more energy than executing the task locally.<sup>4</sup> Cellular networks, for example, can use more energy and incur more delay to offload a task than if the task executed in a WLAN.<sup>3</sup> Future 4G wireless networks and proposed small-cell heterogeneous networks might alleviate this problem because they will have much higher energy efficiency than current cellular architectures.

Recent developments in modeling multiview encoding efficiency might eventually enable adaptive offloading of encoding tasks. Such models take into account video content, wireless channel quality, and differences in network topologies.<sup>2</sup>

## VIDEO DECODING

Decoding can be accomplished by traditional methods or by transcoding a video from one format to another or rendering new views from existing views.

Of the three operations, transcoding is the most straightforward because even in multimedia networks, display devices often have different decoders, screen

resolutions, and associated wireless networks. Consequently, the server must stream the same video content to different devices in 20 to 30 formats. Transcoding a video in an individual format greatly reduces the data transmitted, which requires less energy from the camera networks.

Integrating cloud computing with traditional decoding operations or rendering methods is less straightforward and might require a clean-slate video encoding and decoding technology.

### Cloud-based rendering

Rendering generates a virtual image or video frame from existing feature information. It is central to free-viewpoint video streaming, since the server may not have the requested view among its available candidate views, and is already burdened with transmitting the original video in all the requested views it does have. For 3D video streaming, transmitting feature information such as a 3D mesh- or point-based scene representation can require network bandwidths up to 1 Gbps, which is outside the realm of current mobile environments.

Rendering video frames using cloud computing offers a potential solution. The idea is to select from the candidate views in the cloud one or more reference views closest to the user-requested view and then to transmit only those selected reference views. The user then reconstructs the new video accordingly. In Figure 3, for example, the server considers two reference views, and, instead of transmitting the color pixel frame for each one, it generates a depth map that contains information about the distance of object surfaces from a viewpoint.<sup>5</sup>

With depth map information, the camera can generate a virtual view by warping the received reference views. Warping consists of mapping each view pixel to the requested view's coordinate system and then copying the pixel value. Figure 3 shows warping for two reference views:  $a'$  warped from  $a$  and  $b'$  warped from  $b$ , with  $a' + b'$  being the fusion of  $a'$  and  $b'$ .<sup>5</sup> Fusing two warped views compensates for any occlusion artifact (red ellipses), which occurs when part of the scene's surface appears in one view but cannot be seen in other views.

To select a reference view, the server considers the user's view-changing statistics and opts for a view that maximizes the probability of rendering the requested view within a predefined warping error threshold. That is,

$$\text{Maximize Prob}[rend\_err(V_{r_1}, V_{r_2}, V_{req}) \leq \epsilon]$$

$$V_{r_1}, V_{r_2} \in V_{\text{cld}},$$

where  $V_{r_1}$  and  $V_{r_2}$  are the two reference views,  $V_{\text{cld}}$  is the available set of views,  $rend\_err$  is the warping error, and  $\epsilon$  is the predefined threshold.

Cloud optimization can minimize reference view updates, which helps avoid extra response delay. That is,

$$T_{\text{delay}} = T_{\text{proc}} + \frac{R}{BW} + rtt,$$

where  $T_{\text{proc}}$  is the cloud processing time from the moment the server receives an update request,  $R$  is rate budget,  $BW$  is available bandwidth, and  $rtt$  is the round-trip time. In one optimization scheme,<sup>5</sup> the cloud server predicts the users' watching behavior. Another scheme<sup>6</sup> reduces response delay further by rendering in the cloud and on the client side simultaneously and maximizes reconstructed video quality by jointly optimizing the rate-budget allocation between source and channel coding and between texture and depth frame coding.

### Challenges and possible solutions

When broadcasting the same 3D video in free-viewpoint, different watching preferences can create complex couplings among users, views, textures, depth, and the underlying wireless networks. Optimization in such scenarios is challenging, particularly since most existing research still focuses on single-user cases. Recent work in performance analysis and modeling for adaptive 3D video rendering could provide some solutions.<sup>7,8</sup>

Reducing response delay is another concern. A possible solution is to use a cloudlet server, which essentially moves computations to the edge of the wide area networks. The round-trip delay then becomes much lower than that with remote cloud servers.<sup>9</sup>

### SCALABLE VIDEO NETWORKING

Unlike single users, users of video networking applications must share limited transmission resources in parallel during sessions. To leverage the efficiency of multimedia cloud computing, the cloud server might record and even predict each session's video content characteristics, bandwidth variability, and users' watching preferences and frequency of generating video content. The server must then schedule accordingly so as to optimize video streaming operations.

### 2D video

Scalable video streaming has long been a part of 2D video streaming networks. Streaming optimization schemes adapt frame rate, resolution, or quantization steps to varying network bandwidth. Traditional video encoding standards such as MPEG-4 and H.264/AVC encode a video sequence into several profiles: a primary profile that can be decoded independently to reconstruct basic video quality, and various dependent profiles to enhance that quality.

The prevalent HTTP Adaptive Streaming (HAS) framework suggests delivering streaming video services through



a cloud-client architecture that splits a video sequence into a series of video chunks. The cloud server encodes and transcodes each chunk into several versions with different rates. It then adapts streaming rates to bandwidth by dynamically selecting the streaming rate and a requesting frequency profile for each user.<sup>10</sup>

### 3D video

3D video streaming adds dimensions of scalability to 2D schemes, such as quality and complexity. For example, the cloud server can dynamically adjust the number of views streamed to users (quality) and encode the primary and auxiliary views with different granularity (complexity). It can also apply the scalable strategies of 2D video to different views and to the texture and depth frames for each view.

Considering influencing factors in tandem can increase scalability. In content-aware scalability for multiview video streaming,<sup>11</sup> the idea is to ensure that the cloud server always delivers salient and perceptually sensible visual data with high accuracy even over weak wireless channels or congested networks. A visual attention model analyzes the visual features of each video view jointly, simultaneously considering shape, motion, color, and depth characteristics. The server uses the results of this analysis to ascertain the content's importance level and optimize streaming strategies.

### Networking opportunities and challenges

Recent developments in wireless transmission and networking show promise in furthering scalability in cloud-enabled video streaming. In one approach,<sup>12</sup> the cloud server uses network coding to combine video packets and transmits the combined packets over cellular links (3G and 4G) to smartphones.

The smartphones can receive different sets of packets, which they then exchange ad hoc with each other through Wi-Fi or Bluetooth links. Whenever a smartphone receives a certain number of combined packets, the original video packets can be successfully decoded. Since the smartphone users are considered to be much closer to each other than to the cloud server, packet transmissions among the users are consequently more energy efficient than over cellular links.

Experiments show that each phone saves up to 73 percent of its battery relative to video streaming over cellular networks only. However, the approach is based on the assumption that all smartphones retrieve the same 2D video content. The problem becomes more complex when video content differs, when 3D video streaming must be scaled, or if the framework requires other advanced transmission and networking technologies. In these cases, the cloud server's computational capabilities must accommodate complex optimization problems, which might be outside the resources of a cloudlet.

### COMPRESSIVE SAMPLING

The approaches described so far rely on prediction-based video encoders. Drawbacks include high encoding complexity and choppy video streaming from the variable-length encoding scheme that prediction-based encoders use. The resulting all-or-none behavior creates a synchronization loss in decoding a noise-corrupted packet. This tendency is troublesome in wireless applications, which can have varying levels of noise.

Compressive sampling offers a possible solution to both problems, enabling low-complexity video encoding and error-resilient video streaming.<sup>13</sup> Our work to develop an architecture based on compressive sampling focuses on applications in joint multiview video decoding.

### Multiview streaming

The main idea in compressive sampling is to efficiently acquire and reconstruct signals that are inherently sparse or compressible. Theoretically, an algorithm based on compressive sampling can reconstruct sparse signals with fewer samples than twice the signal's band limit. According to the foundational Shannon-Nyquist signal-processing theory, a perfect reconstruction of a band-limited signal is possible from a countable sample sequence only if the sampling rate is no less than twice the signal's band limit. Compressive sampling successfully breaks that limit.

This lower number of required samples means that scene capture need not involve all the original pixels; thus, encoding reduces to linear operations and most remaining complexity shifts to decoding on the cloud server, ultimately prolonging the camera's battery life.<sup>13</sup> Figure 4 depicts the independent encoding of multiview 3D video sequences based on compressive sampling, along with joint decoding that exploits inter-view correlation, which takes place in the cloud server.<sup>14</sup>

As the figure shows, the camera selects a K-view to serve as a reference for the other camera views—that is, the compressive sampling views. The camera encodes and transmits video frames of these views at a lower measurement rate than it uses for the selected K-view. To decode these compressed views, the cloud server generates side-frame information, first downsampling the K-view frames to the same lower measurement rate. To estimate motion between views, the server compares the reconstructed degraded K-view frames to the initially reconstructed compressed view frame. It then fuses the original compressed view measurements with the generated side image to reconstruct the compressive sampling view frame.

Experimental results<sup>14</sup> show that, with a measurement rate of 0.2 for the compressive sampling view and 0.6 for the K-view, joint decoding yields a 0.85 similarity between the original and decoded compressive

sampling view (a higher similarity corresponds to better reconstruction quality). Independent decoding yields a similarity of only 0.75.

### Challenges and possible solutions

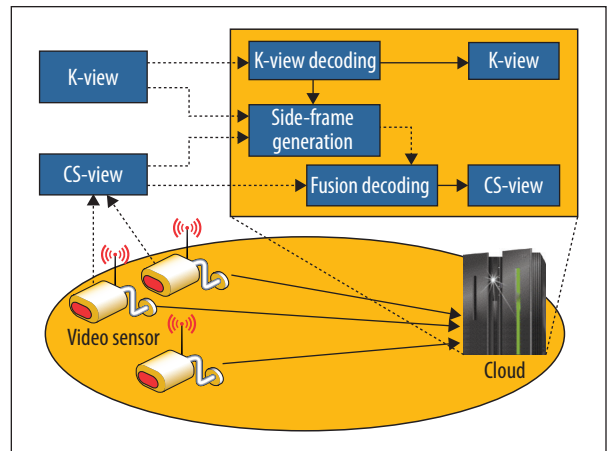
Although an architecture based on compressive sampling has distinct advantages for video capture and encoding with simple operations, the video compression efficiency is still much lower than traditional encoders, which are based on complex motion compensation.

We envision several potential ways to extensively exploit compressive sampling's advantages:


- Develop more efficient and robust decoding algorithms. Such algorithms can raise video reconstruction quality without increasing measurement rate.
- Integrate with traditional encoding to balance efficiency and complexity. An example is to use traditional encoders to encode key frames or views, while using compressive sampling to encode auxiliary frames or compressive sampling views.
- Design network-friendly streaming protocols, by exploiting the error resilience of compressively sampled video signals. This strategy can help improve network-wide energy efficiency, particularly when interference or poorly located cell edges degrade link quality.

Researchers have made progress in the first area, producing algorithms such as least absolute shrinkage and selection operator (LASSO), gradient projection for sparse reconstruction (GPSR), and orthogonal matching pursuit (OMP). In general, however, researchers must still investigate the net benefits of applying compressive sampling in complex scenarios.

**A**lthough cloud-enabled smart camera networks are a promising way to integrate emerging multimedia cloud computing techniques and 3D wireless video streaming, much work remains. Architectures must incorporate energy-efficient task offloading, which requires modeling and predicting energy consumption for both local and cloud computing. Reducing the client-to-cloud delay will enable time-sensitive 3D video streaming. Integrating future heterogeneous wireless networks with multimedia cloud computing will boost emerging multimedia-rich applications. Jointly exploiting the advantages of traditional video encoding technology and compressive sampling could produce clean-slate networking protocols. Finally, integrating solutions will enable researchers to assess the overall benefit of cloud-assisted 3D video networking and better refine it to meet a growing user demand.



**Figure 4. Multiview streaming based on compressive sampling (CS).** The camera selects a K-view, which serves as a reference for the other views (the compressive sampling view). The decoding of both views takes place in the cloud server. Relative to independent decoding, joint decoding yields a higher similarity between the original and reconstructed compressive sampling views.

As a next step in our work, we plan to develop a cloud-assisted video streaming testbed, supporting multiple concurrent free-viewpoint viewers. The testbed can be used to validate newly designed streaming protocols. Through cloud-enabled smart camera networks, we envision that high-quality multiview 3D video streaming will be supported without considerably increasing the energy consumption of smart cameras. 

### Acknowledgments

Work reported in this article is supported in part by the US Office of Naval Research under grant N00014-11-1-0848 and by the US National Science Foundation under grant CNS1117121. Guan's work is also supported in part by the NSFC under grant 61101120 and by the China Postdoctoral Science Foundation under grant 2012M521332.

### References

1. P. Merkle et al., "Stereo Video Encoder Optimization for Mobile Applications," *Proc. 3DTV Conf. True Vision—Capture, Transmission, and Display of 3D Video* (3DTV-CON 11), 2011, pp. 1–4.
2. S. Colonnese, F. Cuomo, and T. Melodia, "An Empirical Model of Multiview Video Coding Efficiency for Wireless Multimedia Sensor Networks," *IEEE Trans. Multimedia*, vol. 15, no. 8, 2013, pp. 1800–1814.
3. Y. Zhao et al., "CAME: Cloud-Assisted Motion Estimation for Mobile Video Compression and Transmission," *Proc. ACM Int'l Workshop Network and Operating System Support for Digital Audio and Video* (NOSSDAV 12), 2012, pp. 95–100.

4. K. Kumar and Y.-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?," *Computer*, vol. 43, no. 4, 2010, pp. 51–56.
5. S. Shi et al., "A High-Quality Low-Delay Remote Rendering System for 3D Video," *Proc. Int'l Conf. Multimedia* (MM 10), 2010, pp. 601–610.
6. D. Miao et al., "Resource Allocation for Cloud-Based Free Viewpoint Video Rendering for Mobile Phones," *Proc. ACM Int'l Conf. Multimedia* (MM 11), 2011, pp. 1237–1240.
7. S. Ma, S. Wang, and W. Gao, "Low Complexity Adaptive View Synthesis Optimization in HEVC-Based 3D Video Coding," *IEEE Trans. Multimedia*, vol. 16, no. 1, 2014, pp. 266–271.
8. F. Shao et al., "Joint Bit Allocation and Rate Control for Coding Multiview Video plus Depth-Based 3D Video," *IEEE Trans. Multimedia*, vol. 15, no. 8, 2011, pp. 1843–1854.
9. T. Verbelen et al., "Cloudlets: Bringing the Cloud to the Mobile User," *Proc. ACM Int'l Workshop Mobile Cloud Computing & Services* (MCS 12), 2012, pp. 29–35.
10. S. Colonnese et al., "Cloud-Assisted Buffer Management for HTTP-Based Mobile Video Streaming," *Proc. ACM Int'l Symp. Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks* (PE-WASUN 13), 2013, pp. 1–7.
11. E. Ekmekcioglu et al., "Content-Aware Delivery of Visual Attention-Based Scalable Multiview Video over P2P," *Proc. IEEE Int'l Packet Video Workshop*, 2012, pp. 71–76.
12. M.R. Zakerinasab and M. Wang, "A Cloud-Assisted Energy-Efficient Video Streaming System for Smartphones," *Proc. IEEE/ACM Int'l Symp. Quality of Service* (IWQoS 13), 2013, pp. 1–10.
13. S. Pudlewski and T. Melodia, "A Tutorial on Encoding and Wireless Transmission of Compressively Sampled

Videos," *IEEE Comm. Surveys and Tutorials*, vol. 15, no. 2, 2013, pp. 754–767.

14. N. Cen, Z. Guan, and T. Melodia, "Joint Decoding of Independently Encoded Compressive Multiview Video Streams," *Proc. Picture Coding Symp. (PCS 13)*, 2013, pp. 341–344.

**Zhangyu Guan** is a postdoctoral research fellow in the Department of Electrical Engineering at the State University of New York at Buffalo. His current research interests are wireless network modeling and optimization, multimedia wireless networks, and mobile cloud computing. Guan received a PhD in communication engineering from Shandong University, Jinan, China. He is a member of IEEE, the IEEE Computer Society, and ACM. Contact him at [zguan2@buffalo.edu](mailto:zguan2@buffalo.edu).

**Tommaso Melodia** is an associate professor in the Department of Electrical Engineering at the State University of New York at Buffalo and director of the university's Wireless Networks and Embedded Systems Laboratory. His research interests include the modeling, optimization, and experimental evaluation of wireless networks with applications to cognitive and cooperative networking, ultrasonic intrabody area networks, multimedia sensor networks, and underwater networks. Melodia received a PhD in electrical and computer engineering from the Georgia Institute of Technology. He is a member of IEEE and on the editorial board of IEEE Transactions on Mobile Computing, IEEE Transactions on Wireless Communications, IEEE Transactions on Multimedia, and Computer Networks. Contact him at [tmelodia@buffalo.edu](mailto:tmelodia@buffalo.edu).



Selected CS articles and columns are available for free at <http://ComputingNow.computer.org>.

## Silver Bullet Security Podcast

In-depth interviews with security gurus. Hosted by Gary McGraw.



[www.computer.org/security/podcasts](http://www.computer.org/security/podcasts)

\*Also available at iTunes

Sponsored by