Share Risk and Energy: Sampling and Communication Strategies for Multi-Camera Wireless Monitoring Networks

Zichong Chen[†], Guillermo Barrenetxea[†], and Martin Vetterli[†‡]

[†]School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
[‡]Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720, USA

Abstract—In the context of environmental monitoring, outdoor wireless cameras are vulnerable to natural hazards. To benefit from the inexpensive imaging sensors, we introduce a multi-camera monitoring system to share the physical risk. With multiple cameras focusing at a common scenery of interest, we propose an interleaved sampling strategy to minimize per-camera consumption by distributing sampling tasks among cameras. To overcome the uncertainties in the sensor network, we propose a robust adaptive synchronization scheme to build optimal sampling configuration by exploiting the broadcast nature of wireless communication. The theory as well as simulation results verify the fast convergence and robustness of the algorithm.

Under the interleaved sampling configuration, we propose three video coding methods to compress correlated video streams from disjoint cameras, namely, distributed/independent/joint coding schemes. The energy profiling on a two-camera system shows that independent and joint coding perform substantially better. The comparison between two-camera and single-camera system shows 30%-50% per-camera consumption reduction. On top of these, we point out that MIMO technology can be potentially utilized to push the communication consumption even lower.

I. INTRODUCTION

Wireless sensor networks find widespread applications in environmental monitoring, where use cases range from wildlife monitoring [9] to microclimate monitoring [2]. Thanks to the availability of low-cost image sensor chips, such as CMOS cameras, wireless sensor networks can extend their functionalities to include image and video monitoring [7]. Among the many applications of cameras in environmental monitoring, we are especially interested in natural hazards detection, such as avalanche. To detect such a random event, a single monitoring camera is usually programmed to capture the scenery of interest periodically (1-60 images/hour in our case), and transfer the image sequence back to a base station (BS) in real-time. However, the camera itself is also experiencing unpredictable weather conditions and consequently can be affected by surrounding major events. For example, Fig. 1 shows a wireless sensor station destroyed by an avalanche in a recent deployment of ours.

Moreover, for visual applications to become ubiquitous, we need to tackle the problem of processing and transmitting large amounts of data, under the severe energy constraints of wireless sensor networks (e.g., nodes operating on batteries
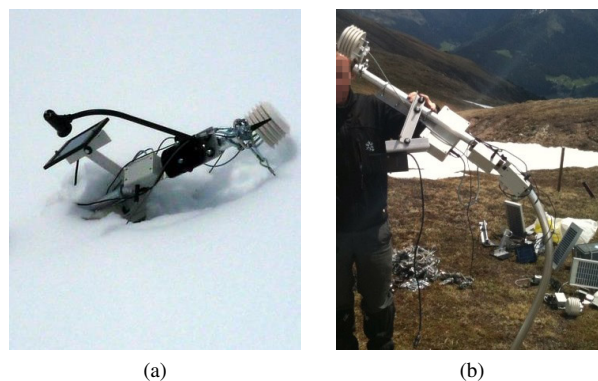

Fig. 1. Environmental monitoring hazards in the Swiss Alps. A SwissEx (http://www.swiss-experiment.ch) wireless sensor station at Davos destroyed by an avalanche: (a) The station after the avalanche. (b) After snow melt.

and a solar panel). As the radio transceiver is often the largest energy consumer, minimizing the transmitted data is a necessary step to augment wireless sensor networks with image/video capabilities.

In this paper, we show that a multi-camera wireless network provides a solution to the two main concerns of visual monitoring: (i) It increases robustness to unpredictable events by adding redundancy. (ii) It allows to reduce energy consumption on each camera by distributing image sampling tasks among all neighboring cameras, while the event detection probability is kept the same.

In order to increase robustness, we consider the scenario where cameras are in different locations but observe a common scenery of interest. Therefore, they generate correlated image sequences (video). In case of camera failure, as long as there is still one surviving, we will not lose the visual access to the scenery of interest. Meanwhile, surviving cameras also have a good chance to record the invaluable event that destroyed other cameras (e.g., an avalanche), and can transfer the images back to the base station.

In normal operations, to deliver all images generated by such a multi-camera system, each camera communicates with the BS through a wireless link (we consider long-range communication scenario). Based on this configuration, we propose

a general image capture (sampling) framework to solve two problems raised in multi-camera system:

- From an event detection point of view, what is an optimal sampling strategy using multiple cameras?
- Since views from disjoint cameras have strong correlations, what is an energy efficient way to deliver images from all cameras to the BS?

Motivated by the successive refinement coding of two static sources [4], we extend this scheme to encode videos[1] from n cameras: each camera samples at frequency f/n, and the sampling time of cameras are arranged in an interleaved manner, imitating a single camera sampling at frequency f. In this way, we distribute the risk of physical hazard and energy burden of wireless transmission into n independent cameras, while the overall event detection capability is kept the same.

In this paper, we develop this idea and show that a uniform sampling configuration is optimal in terms of event detection probability. To tackle the unknown clock offsets and possible camera failure, we propose a robust adaptive synchronization algorithm to build the optimal sampling structure that exploits the broadcast nature of wireless communication. In case of new camera arrival or camera malfunction, cameras autonomously adapt to the optimal configuration.

To compress all image sequences from n disjoint cameras, we use distributed video coding (DVC) schemes [6] [12]. Alternatively, we propose a joint video coding scheme by exploiting the fact that for long-range communication, energy consumption is significantly lower in reception (RX) than in transmission (TX). The final energy profiling on a two-camera system shows that joint coding offers a better compression rate compared to the distributed coding scheme.

Our main contributions in this paper are:

1) We propose a novel sampling framework to fully exploit the advantage of multiple cameras, which can distribute risk and energy among n cameras, and the consumption per camera is reduced by a factor of n.
2) We find that a joint coding method outperforms the state-of-art DVC scheme substantially in our application, where sampling rate is much lower than conventional video and thus inter-frame correlation is not strong enough for DVC to be efficient.
3) We point out the potential of MIMO technology: a multi-camera network has fundamental energy saving as compared with traditional single-camera system, and thus in principle the communication consumption per camera is reduced by a factor of n^2.

The remainder of this paper is organized as follows: We present related works in Section II. In Section III, we introduce an optimal sampling framework for multiple cameras, and propose a robust adaptive synchronization scheme to build the correct sampling structure autonomously. Simulation results are also given to verify this algorithm. Section IV introduces

[1] Note that for environmental monitoring applications the frame rate is much lower than conventional video at 30 fps.

three video coding methods to compress image sequences from all cameras, namely, distributed/independent/joint coding schemes. Section V evaluates a two-camera system, where the experimental setup and detailed results are given. Finally, in Section VI we point out the potential of MIMO technology for multi-camera systems.

II. RELATED WORK

Low power and inexpensive wireless imaging sensors [13] have became available in recent years. They were originally intended for gesture recognition, indoor monitoring, and object retrieval [10] [17]. Wireless cameras used for environmental monitoring have also been proposed [7], but cameras are treated as independent nodes. Our work on a multi-camera wireless monitoring network addresses the following two issues: (i) sampling by multiple sensors; (ii) distributed video coding.

The general idea to exploit the sampling capability of multiple sensors is inspired by space-time sound field sampling theory [1], where the optimal sampling grid is achieved by a tight packing of sound field's spectrum replicas. We translate this idea to the field of camera networks, where multiple cameras monitor a common scenery of interest for robustness purposes. By interleaving the sampling time of cameras, we find the optimal sampling configuration to maximize the event detection probability. This is also related to our previous work on successive refinement coding of two static sources [4] [5], where two cameras cooperate in a Ping-Pong fashion to achieve joint successive refinement coding of stereo-view images. Our proposed scheme in this paper can be thought as an extension from two cameras to multiple cameras, as well as from static images to video streams.

Compressing interleaved multi-view videos is related to multi-view distributed video coding. This area has been rapidly evolving recently in the context of conventional videos, due to standardization efforts for 3DTV. As opposed to our setup, multi-view distributed coding tackles the non-interleaved setup. The most recent works of this area include homography/epipolar based schemes [14], and PRISM derived schemes [18] which was originally developed as single-view coding scheme [12].

III. OPTIMAL SAMPLING USING MULTIPLE CAMERAS

In this section, we introduce an interleaved sampling structure for a multi-camera network. It is designed to distribute physical risk and energy burden among all deployed cameras, while being optimal in terms of event detection probability. Then, we propose a robust adaptive synchronization scheme, that can autonomously learn the network configuration and converge exponentially to the optimal sampling structure.

A. Interleaved sampling structure

Suppose we have some wireless cameras equipped with long-range radio (typically several kilometers, up to 20km), and each can be programmed to capture images periodically. Initially we install one camera to monitor a scenery of interest,
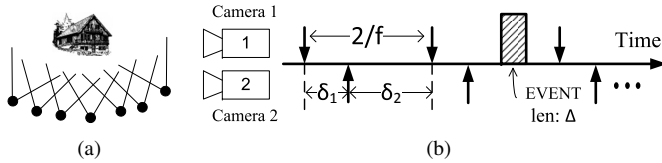
Fig. 2. Multi-camera monitoring network: (a) By monitoring a common scenery of interest, multiple cameras (represented by dots) share the risk of physical hazard and energy burden of wireless transmission. (b) Interleaved sampling setup of two cameras trying to capture a random event.

Algorithm 1 Adaptive synchronization algorithm

1: **if** camera startup **then**
2:   Wait for $T = n/f$
3:   **if** detect a beacon **then**
4:     Set a random time to initiate the first TX
5:   **else**
6:     I'm *Anchor*, initiate a TX immediately
7:   **end if**
8: **else if** I'm *Anchor* **then**
9:   Repeat TX every T
10: **else**
11:   x_k, x_{k+1}: current/next TX schedule of "me"
12:   x^-, x^+: time-stamp of two adjacent beacons to x_k
13:   $x_{k+1} = T + (x^- + x^+)/2$
14:   **if** haven't received *Anchor* beacon for $M \cdot T$, $(M > 1)$ **then**
15:     I'm *Anchor*, initiate a TX immediately
16:   **end if**
17: **end if**
18: $k = k + 1$

and it captures images at a fixed sampling frequency f. In order to increase the robustness of system, we deployed $n-1$ extra cameras to monitor the same scenery of interest (see Fig. 2a). In order to share the energy burden among all cameras, we reduce the sampling frequency of each camera to f/n. Under such a scenario, we want to find the optimal sampling strategy that maximizes the probability to catch/record an event.

Fig. 2b shows a two-camera setup as described above: the arrows represent the time instants of captured images. The sampling frequency of each camera is fixed at $f/2$, and δ_1 is the interleaved offset between the sampling grid of two cameras. If we assume a particular event happening randomly (uniform distribution) with a time length Δ, then with a sampling scheme like in Fig. 2b, the probability P_d that we catch the event is related to the choice of δ_1. In the following, we prove that the optimal sampling strategy is a symmetric interleaved setup.

Theorem 1: For n cameras each sampling at frequency f/n, the optimal sampling strategy is a symmetric interleaved setup with uniform spacing of $1/f$.

Proof: Denote $\delta_1, \delta_2, \cdots, \delta_n$ be the interleaved offsets between the sampling grid of adjacent cameras (In Fig. 2b, the case $n = 2$ is shown). They are constrained by $\sum_{i=1}^{n} \delta_i = n/f$.

When $n = 2$ (we can assume $\delta_1 < 1/f$ because of symmetry), since the event occurs at a uniform distribution with a time length Δ, the probability P_d that we catch the event can be expressed as

$$P_d = \begin{cases} f\Delta, & \Delta < \delta_1 \\ f(\Delta + \delta_1)/2, & \delta_1 < \Delta < 2/f - \delta_1 \\ 1, & \Delta > 2/f - \delta_1 \end{cases}.$$

Thus, the optimal choice of δ_1 that maximizes the detection probability is

$$\arg\max_{\delta_1} P_d = \begin{cases} \delta_1 > \Delta, & \Delta < 1/f \\ \delta_1 < \Delta, & \Delta > 1/f \end{cases},$$

from which we know that $\delta_1 = 1/f$ is the universal choice to maximize P_d. Therefore, in the optimal sampling setting, two cameras are sampling in a symmetric interleaved manner: each samples at frequency $f/2$, and the interleaved offset is $1/f$.

We can generalize this to the case when $n > 2$: For δ_k and δ_{k+1}, when the random event happens inside them, the optimal strategy is to choose $\delta_k = \delta_{k+1}$ according to the argument of two-camera case above.

By doing such partition from δ_1 to δ_{n-1}, we can get $\delta_1 = \delta_2$, $\delta_2 = \delta_3$, $\cdots$, $\delta_{n-1} = \delta_n$. Thus the symmetric interleaved setup $\delta_1 = \delta_2 = \cdots = \delta_n$ is the optimal sampling strategy. ∎

B. Adaptive synchronization scheme

Knowing the optimal sampling structure of a multi-camera network, it is of practical interest to investigate if these cameras can autonomously setup such a configuration, given the **uncertainties** presented in sensor networks:

- n decreases: some camera fails to work.
- n increases: a new camera is installed.
- Local clocks of cameras are not synchronized and are subject to drift.

In the following, we propose an algorithm to tackle all these problems. We make some practical assumptions: (i) All cameras are equipped with long-range radio to communicate with the BS, and they are able to overhear each other (all within the radio communication range); (ii) Every camera has a timer, and they send its own image periodically at a fixed interval $T = n/f$.

We denote the first installed camera as an *Anchor* camera, which fixes its own schedule irrespective of other cameras. Algorithm 1 illustrates the detailed iterative scheme to achieve adaptive synchronization. The basic idea is that each camera sends a beacon followed by its own image (referred to as a TX) according to the scheduled interval T. Meanwhile, each camera records the time-stamps of the two adjacent beacons to its current transmission and adjust its next transmission schedule to be located at the same distance from the predicted
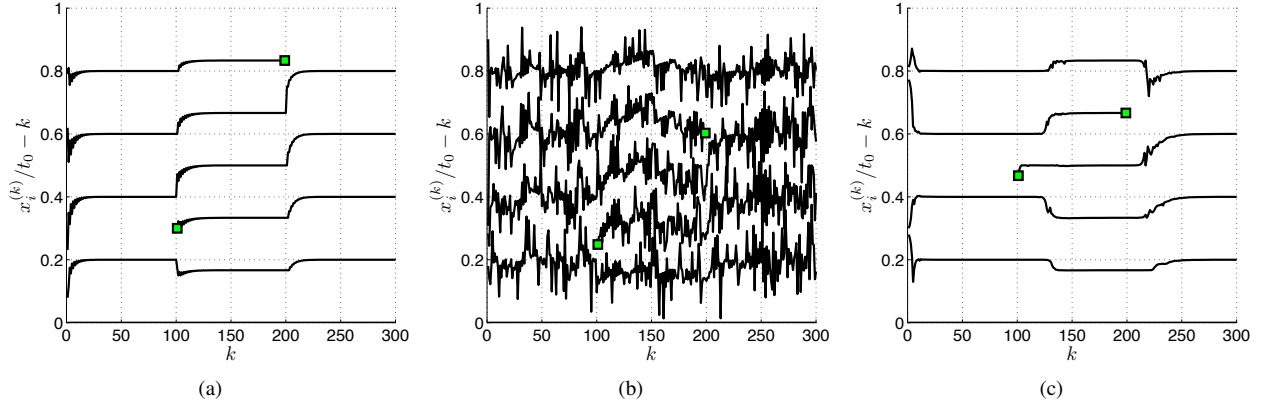
Fig. 3. Simulation results of adaptive synchronization scheme. Five unsynchronized cameras starts at $k = 0$, one camera pops up at $k = 100$, and one random camera dies at $k = 200$. The green squares labels the camera that shows/disappears. k denotes the number of sampling intervals passed. $x_i^{(k)}/T - k$ denotes the normalized sampling time of camera i. (a) Algorithm 1 under perfect overhearing condition. (b) Algorithm 1 under 10% overhearing loss. (c) Algorithm 1 improved by robust update (Algorithm 2) under 10% overhearing loss.

two adjacent beacons. In this way, when some cameras suddenly show up or disappear, each camera can adaptively adjust the transmission schedules to recover the correct sampling configuration. Note that it is also possible that the *Anchor* will fail to work in the long run, thus it is important to identify this camera with a special beacon so that others can claim a new *Anchor* when they found there is no *Anchor* alive.

In practice, the camera number n is actually not necessary to any nodes. Once the system is alive, each node sticks to their original sampling interval. When new camera is installed, they automatically adapt to the new sampling configuration, and thus the overall sampling frequency of the system is increased. On the other hand, since n is known at the initial deployment, we can reduce the sampling frequency of all cameras, and thus the designed power budget of each camera (solar panel size) can be reduced.

We prove the convergence of Algorithm 1 in the following:

Theorem 2: Under the condition of perfect overhearing, the adaptive synchronization algorithm converges to the symmetric interleaved sampling structure in an exponential manner.

Proof: Denote the number of cameras as n, the scheduled interval of each camera as T, and the time-stamp of camera i's kth TX as $x_i^{(k)}$, $i = 1, \ldots, n$, $k = 1, 2, \ldots$. Under the condition of perfect overhearing, any camera can hear other cameras' beacons without any loss. Therefore, we can interpret the adaptive synchronization algorithm as:

$$X^{(k+1)} = A \cdot X^{(k)} + \eta, \tag{1}$$

where $X^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \cdots, x_{n-1}^{(k)}, x_n^{(k)}\right)^{\mathrm{T}}$, and

$$A = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & \ddots & \vdots \\ 0 & \frac{1}{2} & 0 & \ddots & 0 \\ \ddots & \ddots & \ddots & \ddots & \frac{1}{2} \\ \frac{1}{2} & 0 \cdots & 0 & \frac{1}{2} & 0 \end{pmatrix}, \ \eta = \begin{pmatrix} T \\ T \\ T \\ \vdots \\ T \\ \frac{3T}{2} \end{pmatrix}.$$

$$|A - \lambda I| = \frac{(1-\lambda)}{\sqrt{\lambda^2-1}}\left(\left(\frac{-\lambda+\sqrt{\lambda^2-1}}{2}\right)^n - \left(\frac{-\lambda-\sqrt{\lambda^2-1}}{2}\right)^n\right),$$

thus the eigenvalues of A are

$$\lambda_i = \cos\frac{(i-1)\pi}{n}, \ i = 1, \cdots, n.$$

Given the structure of A, the corresponding eigenvector v_i satisfies $v_1 = (1, 1, \cdots, 1)^{\mathrm{T}}$, and $v_i = \left(0, v_i^{(2)}, \cdots, v_i^{(n)}\right)^{\mathrm{T}}$ for $i \neq 1$. Thus, we can expand

$$X^{(1)} = x_1^{(1)}v_1 + \sum_{i=2}^{n}\alpha_i v_i, \ \eta = Tv_1 + \sum_{i=2}^{n}\beta_i v_i,$$

which are then plugged into the solution of recurrence (1):

$$X^{(k)} = A^k X^{(1)} + \sum_{j=0}^{k-1}A^j\eta$$

$$= x_1^{(1)}\lambda_1^k v_1 + \sum_{i=2}^{n}\alpha_i\lambda_i^k v_i + \sum_{j=0}^{k-1}\left(T\lambda_1^j v_1 + \sum_{i=2}^{n}\beta_i\lambda_i^j v_i\right)$$

$$= (x_1^{(1)} + kT)v_1 + \sum_{i=2}^{n}\left(\alpha_i\lambda_i^k + \beta_i\frac{1-\lambda_i^k}{1-\lambda_i}\right)v_i.$$

Since $|\lambda_i| < 1$ for $i > 1$, $X^{(k)} - (x_1^{(1)} + kT)v_1$ converges exponentially to a unique solution

$$\lim_{k\to\infty}\left(X^{(k)} - (x_1^{(1)} + kT)v_1\right) = \sum_{i=2}^{n}\frac{\beta_i}{1-\lambda_i}v_i.$$

On the other hand, it is straightforward to verify that $X^{(k)} - (x_1^{(1)} + kT)v_1 = T \cdot \left(0, \frac{1}{n}, \cdots, \frac{n-1}{n}\right)^{\mathrm{T}}$ satisfies the convergence condition. Therefore, the algorithm converges to the symmetric interleaved sampling structure. ∎

Fig. 3 shows the simulation results of the adaptive synchronization scheme. Five unsynchronized cameras starts at $k = 0$, one camera appears with a random sampling time at $k = 100$, and one random camera dies at $k = 200$. Here k denotes the number of sampling intervals elapsed, thus it

Algorithm 2 Robust schedule update algorithm (replace Lines 11-13 of Algorithm 1).

1: x_k, x_{k+1}: current/next TX schedule of "me"
2: x^-, x^+: time-stamp of two adjacent beacons to x_k
3: $x_\Delta = (x^- + x^+)/2 - x_k$
4: **if** $|x_\Delta| > $ thre **then**
5: $\quad x_{k+1} = T + x_k + \text{sgn}(x_\Delta) \cdot$ thre
6: **else**
7: $\quad x_{k+1} = T + x_k + x_\Delta$
8: **end if**
9: **if** $|x_\Delta| > 2 \cdot$ thre **then**
10: $\quad$ thre $= 2 \cdot$ thre
11: **else if** $|x_\Delta| < $ thre$/2$ **then**
12: $\quad$ thre $=$ thre$/2$
13: **end if**

---

represents the kth TX of each camera. $x_i^{(k)}/T - k$ denotes the normalized sampling time of camera i, which should converge to $(i-1)/n$ when there are n cameras (same notations as in the proof of Theorem 2). Each trace records the evolution of a camera's normalized sampling time. Notice that the *Anchor* camera is setup so that it always coincides with the horizontal axis. Fig. 3a shows the performance of Algorithm 1 under perfect overhearing condition. It can be seen that all cameras do converge to the optimal sampling structure quickly (within 25 steps), withstanding all uncertainties as we mentioned in the beginning of Sec. III-B.

However, if the perfect overhearing condition does not hold, Algorithm 1 could degrade badly. Fig. 3b shows the performance when each camera has 10% possibility to miss the beacon sent by other cameras, which suggests that the original algorithm totally fails under packet losses. To overcome this, we propose a robust schedule update algorithm that adaptively limits the change rate of transmission schedule. As described in Algorithm 2, each camera has a threshold variable thre, which is initialized to a certain value (e.g., $0.01 \cdot T$) at startup. At each update round, the maximum adjustment is limited to thre, meanwhile thre is also adaptively adjusted. In this way, the algorithm can learn the optimal threshold, and is robust to some burst errors caused by overhearing loss. Fig. 3b shows the performance of Algorithm 1 with this improvement. As we can see, with the same overhearing loss rate, it converges fast (within 50 steps), and keeps steady after it converges.

IV. VIDEO CODING OF MULTI-CAMERA SYSTEM

Under the optimal interleaved sampling configuration, multiple cameras cooperate to monitor a common scenery. If the images sequences from disjoint cameras are registered properly so that the shared views are aligned, then they actually forms a single video sequence which can be thought of as being captured by a single camera sampling at frequency f (See Fig. 4 as an illustration of the two-camera case). To exploit this fact, we investigate three video coding schemes, namely, distributed/independent/joint coding schemes to compress the
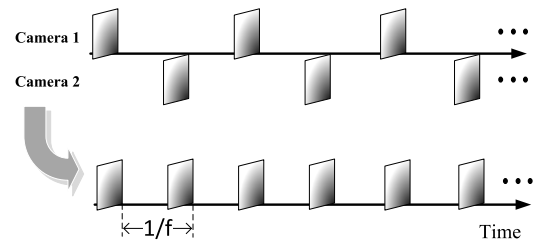

Fig. 4. Interleaved sampling: dual image sequences can be regarded as a single video stream.
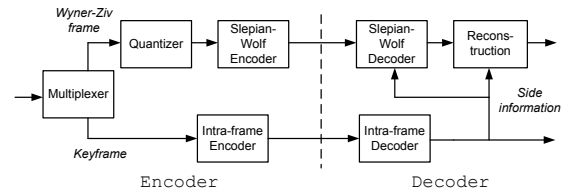

Fig. 5. Block diagram of the DISCOVER encoding/decoding architecture.

data from the cameras. We assume the view registrations between static cameras have been estimated[2], and only the overlapped area is considered in the coding process (non-overlapped areas are encoded independently using conventional methods).

A. Independent video coding

The difficulty in merging/coding image sequences of multiple cameras is that they require direct communication between cameras which is usually not feasible. Therefore, a simple solution to avoid inter-camera communication is to apply a conventional video coder like H.264 [11] independently to each camera.

B. Distributed video coding

Since consecutive frames are from physically separated cameras, distributed video coding (DVC) is a specialized tool recently developed for such compression task [6] [12]. In such schemes, the significant burden of motion estimation is shifted from the encoder to the decoder, and therefore the encoder virtually assumes no knowledge of previous and following frames while encoding a particular video frame.

DISCOVER [6] is one of the mainstream distributed video coders that performs close to the standard video compression method like H.264. The basic idea (see Fig. 5) is to multiplex the input video stream into keyframes and Wyner-Ziv frames, and the keyframe is regarded as the side information to reconstruct the Wyner-Ziv frame at the decoder. The encoder assumes no knowledge of keyframes while encoding the Wyner-Ziv frame, thus no communication is necessary if we distribute keyframes and Wyner-Ziv frames among cameras. The advantage of such a scheme is that it uses a significantly lower encoding complexity than traditional video compression methods, and therefore, it can be easily implemented on an embedded platform consuming little CPU power.
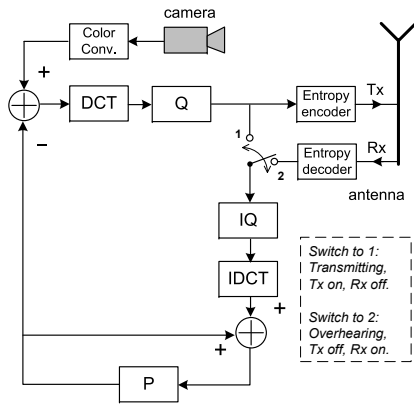
[2]See [3] for image registration algorithm.

Fig. 6. Virtual joint video coding based on a hybrid coder (encoder part at each camera). Q is quantization block, P is prediction block, and Ixx is the inverse operation of xx. The motion data estimated in the prediction block is also sent but not explicitly drawn in the diagram.

C. Joint video coding by overhearing

Since the image sampling rate of environmental monitoring application is constraint by severe energy supply, the radio channel occupation remains very low (see [3] for calculations). For interleaved sampling configurations, this implies that two cameras will not transmit simultaneously and no interference exists. Therefore, the long range radio used between the cameras and the BS (typically 20km) provides a sufficient overhearing coverage for neighboring cameras. On the other hand, energy consumptions of transmission (TX) and reception (RX) are highly non-symmetric especially in the long-range setup (see Table I), this motivates us to use overhearing since the coding gain in TX will compensate the energy loss in RX.

TABLE I
POWER CONSUMPTION OF SHORT AND LONG RANGE TRANSCEIVERS [16].

	GSM	802.11b	Bluetooth
RX (mW)	240	60	30
TX (mW)	2860	350	14.5
Range (m)	20k	40	10

In contrast with the distributed coding approach, we propose a joint video coding scheme that uses overhearing as a passive link between cameras. Fig. 6 shows a modified H.264 hybrid coder to achieve joint coding of all cameras. Each camera has such an encoding unit, in which the feedback loop of the hybrid coder is split into two different routes. The first route (when switch=1) is identical to a conventional hybrid coder, where the source image is fed into the prediction block. On the contrary, the second route (when switch=2) bypasses the overheard message into the prediction block, which is the codewords generated by other camera.

Each camera switches to 1 when it is in transmission mode, otherwise it switches to 2 for overhearing. By repeating this cooperation, the prediction blocks of all cameras are always synchronized and store the latest frame for joint coding. Therefore, the encoder at each camera is virtually compressing

a video sampled at frequency f, rather than its own sampling frequency f/n. As a consequence, inter-camera correlation is fully exploited and better compression ratio is expected.

V. CASE STUDY: ENERGY PROFILING OF A TWO-CAMERA SYSTEM

The most critical factor in evaluating a wireless camera network is the energy consumption[3]. To get an approximate energy profiling of a two-camera system, we first collect some image datasets for benchmarking purposes using two conventional cameras. Then we can obtain the computation consumption by running the previous described video compression schemes using these datasets on a typical embedded image processing platform *Sensorcam* [3]. Meanwhile, the video size after compression can be used to estimate the communication consumption by using typical long-range radio power (see Table I). Then, the global energy profiling is the summation of computation and communication consumptions. In the following, we first introduce the experimental setup, and then give the detailed evaluation results.

A. Experimental setup

Fig. 7a shows the two monitoring cameras deployed on the roof of a building used to capture benchmark images. Due to the limited space, two cameras are placed closer than a real-world monitoring setup. Therefore, we choose two different sceneries with different depth structures (see Fig. 7b and Fig. 7c) to investigate the algorithm: Scene A captures the facade of a building which is a planar scene suitable for homography geometry. In contrast, Scene B includes buildings stretching out to the mountains far away, which has a complex depth structure. The later one introduces parallax between two views mimicking two cameras deployed in a distributed manner.

Our image processing platform, *Sensorcam*, includes an ARM based Marvell XScale PXA270 module which runs an embedded Linux system. All algorithms described in Sec. V are implemented using C/C++ for running on the *Sensorcam* board. Refer to [3] for detailed information about datasets and implementation. Table II lists some of the parameters related to the experiments.

B. Compression-complexity tradeoff

The global energy consumed at each camera can be expressed as:

$$E_{\text{global}} = E_{\text{ENC}} + E_{\text{TX}} + E_{\text{RX}}, \qquad (2)$$

where E_{ENC} denotes the computation energy spent on video encoding (e.g., H.264), E_{TX} denotes the communication energy for transmitting codewords, and E_{RX} denotes the communication energy for overhearing. From the global energy point of view, it is worth considering what is the most energy efficient allocation for computation and communication: As the computation complexity increases, the size of bitstream

---

[3]The real-time monitoring does not allow latency, so all video coding schemes we used in this paper send images immediately after captured.
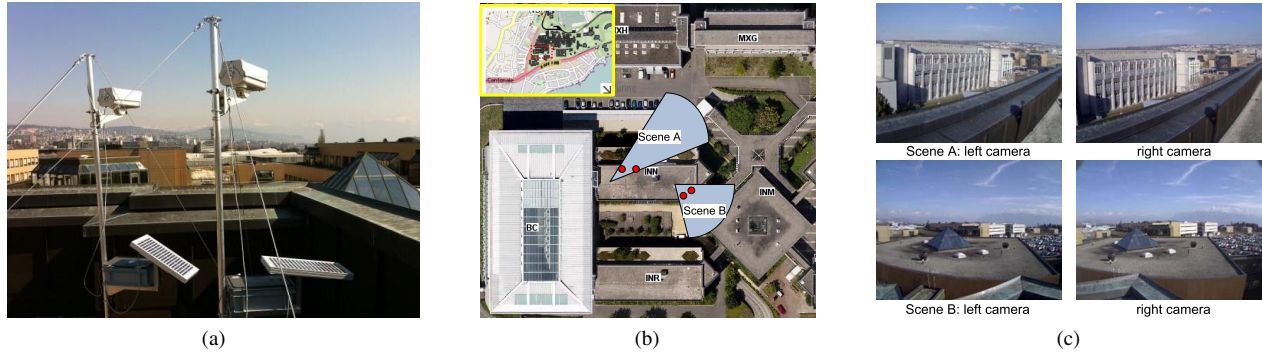
Fig. 7. Experimental setup to capture datasets: (a) Two surveillance cameras with solar power system watching Scene B. (b) The setup of two cameras (red dots) on roof of a building in campus. Scene A and B show the viewpoints of two datasets. (c) The stereo image samples of Scene A and B respectively.

TABLE II
PARAMETERS USED IN ENERGY PROFILING

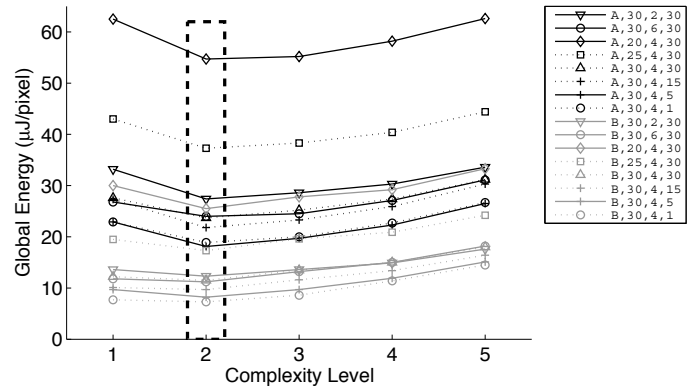| *Sensorcam* [3]	
CPU	520MHz
SDRAM	64MB
FLASH	32MB
Consumption (normal)	800mW
Consumption (sleep)	7mW
Long-range radio [16]	
Consumption (Tx@30dBm)	2860mW
Consumption (Rx)	240mW
Consumption (idle)	14mW
Speed	64Kbps
camera	
Resolution	640×480
Consumption (active)	120mW

Fig. 8. The global energy consumption using H.264 video coder with respect to its complexity level. The legend shows the parameters used: data-set name, QP (target quality control), GOP (group of pictures) and T (sampling interval, equals $1/f$). Each test encodes 12 frames for speed consideration.

after source coding gradually approaches the rate-distortion limit. On the other hand, more computation energy is used to achieve high compression ratio.

The H.264 encoder provides some preset options to tradeoff compression efficiency against encoding speed. For notation purpose, we label these complexity presets as level 1-10 (level 1 has the lowest complexity, level 10 has the best compression ratio). We evaluate the optimal tradeoff, under different target qualities, GOPs (group of pictures[4]), and sampling intervals (inverse of sampling frequency f as defined in Sec. III-A).

Fig. 8 shows the global energy profiles (measured in μJ/pixel) of some representative combinations of three factors for Scene A and Scene B. It is interesting to observe that the optimal choice of H.264 complexity always falls at **level 2**. Therefore, in the following we set the complexity of H.264 encoder to level 2.

C. Video compression comparisons

In Sec. IV we have introduced three different video coding schemes for multi-camera system, including distributed video coding (DISCOVER), H.264 based joint coding scheme, and H.264 based independent coding scheme. We now compare these three schemes.

[4]For error-resilience purpose. E.g., if sampling interval is 1min and GOP=4, then we can recover from packet loss within 4min.
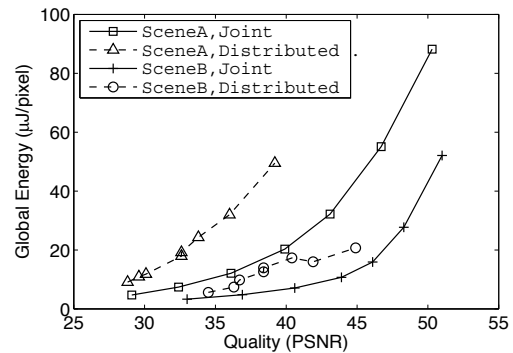

Fig. 9. Distributed versus Joint video coding: global energy versus quality curves of system using distributed/joint video coding for two data-sets. The GOP parameters are chosen to maximize video compression ratio. The sampling interval is fixed to 5min. Each test encodes 30 frames (5 frames for DISCOVER) due to speed consideration.

For all schemes, the global energy consumption can be obtained using (2). Particularly, $E_{RX} = 0$ for independent/distributed coding schemes, and $E_{RX} \neq 0$ for the joint coding scheme since overhearing is required. Fig. 9 shows the comparison between distributed and joint video coding
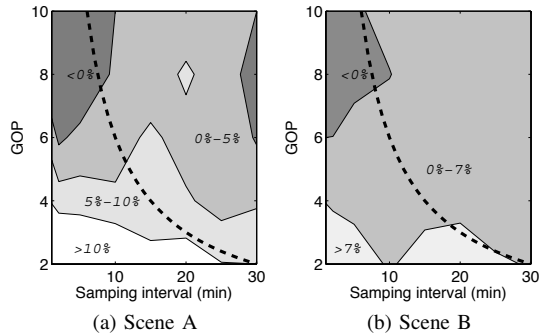
(a) Scene A   (b) Scene B

Fig. 10.   Independent versus Joint video coding: global energy saving (measured in %) of joint coding compared with independent video coding. The saving is plotted in a 2D-parameter space of (sampling interval, GOP), which is interpolated by 64 grid samples and value of each sample is obtained by encoding 32 frames for speed consideration. The target qualities (QP) of both schemes are equal and fixed. The dashed lines show parameter combinations that have a constant error-resilience capability (recover in max. 60min).

schemes using the two datasets. It can be seen that joint video coding outperforms distributed video coding. The significant loss for distributed coding suggests that the state-of-art DVC schemes rely heavily on strong correlations between successive video frames, while this is usually not feasible in an application like environmental monitoring (sampling rate is much lower: $0.0003 - 0.02$ fps versus 30 fps in conventional video coding).

Fig. 10 shows the energy saving of joint coding compared to independent video coding. Notice that since the sampling rate of environmental monitoring is much lower than in regular video coding, the GOP value of interest is also lower. The dashed lines show parameter combinations that have a constant error-resilience capability, namely the system can recover from an error in maximum 60min. Clearly, as the sampling interval increases (traverses on the dashed line from left to right), the advantage of joint video coding appears. Eventually, when the sampling interval goes to 20-30min, the energy saving is close to 10%. To sum up,

1) For environmental monitoring with static scenery, distributed coding fails to exploit the temporal correlation between disjoint cameras.
2) In long-range communication setup, joint coding by overhearing outperforms independent coding when the sampling rate is low.
3) Considering the system complexity, when the sampling rate is not low enough (i.e., sampling interval smaller than 10min), independent coding is the most efficient video compression scheme for multi-camera system.

D. Per-camera consumption

Fig. 11 illustrates the per-camera real-time consumption of a two-camera system using H.264 based joint coding (depicted by solid lines with three consumption categories: TX, RX, and CPU). As we can see, the communication (TX+RX) and computation (CPU) consumptions approximately share 3:1 of the global energy. Moreover, the computation consumption is
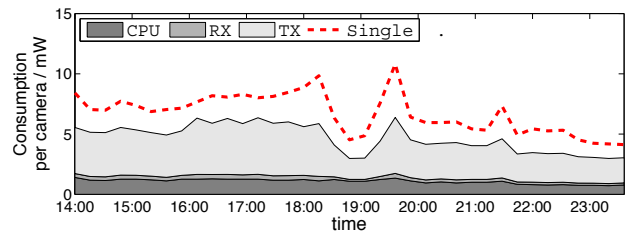

Fig. 11.   Per-camera real-time consumption of a two-camera system using joint coding (solid lines). As a comparison, the dashed line shows the consumption of a single-camera system with the same sampling frequency. The data-set used is Scene A (14:00-24:00), the sampling interval is 4min, and GOP is 4. The consumption of the two-camera system consists of three categories: CPU for computation energy spent on video encoding, TX for energy of wireless transmission, and RX for overhearing. Single denotes the global consumption of single-camera system (including all three categories).
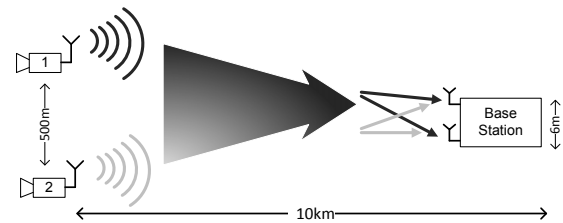

Fig. 12.   How Space-Division Multiple Access (SDMA) works to create MIMO advantage for environmental monitoring camera networks.

generally very static, while the communication consumption varies as the scene changes (e.g., consumes less at night because scene structure is much simpler).

All previous evaluations focus on the two-camera system. We can also evaluate the global consumption of a naive single-camera monitoring system as a comparison. Despite losing the robustness, a single camera does not have overheads like registration error and overhearing consumption. To verify if multiple cameras really share the energy cost, we check *if the per-camera consumption in a multi-camera system is smaller than the consumption in single-camera case?*

The dashed line in Fig. 11 shows the real-time consumption of a single-camera system with the same sampling frequency. The comparison shows that the per-camera consumption of the two-camera system is reduced by 30%-50%, depending on the scenery condition. This energy saving factor is approximately consistent with the factor of camera number, which is 2 in this case.

VI. MIMO ENERGY SAVING

In the previous sections, we have shown that a multi-camera system can distribute the risk and energy burden. Apart from the sampling and coding scheme we present in the previous sections, with the MIMO technology [15], we can push the communication consumption even lower: *Multi-camera system requires less overall communication energy than a single camera.* In practice, MIMO has been successfully applied to indoor application like IEEE 802.11n WLAN. We expect in the near future practical MIMO communication hardware will also be used for wireless sensor networks [8].

The basic principle of 802.11n is to use multiple antennas and a rich scattering multi-path environment to create spatial diversity. Unlike the indoor scenario for 802.11n, the environmental monitoring camera operates in open space where multi-path scattering is much lower. In this case, the uplink MIMO channel from TX antennas to the base station (BS) is a well conditioned fading channel only if the antennas are far apart. Therefore, physically separated wireless cameras provides a potentially good MIMO setup.

Fig. 12 shows an example: Two cameras (each has one TX antenna) are located 10km away from the BS and they use 1GHz band for cameras-to-BS communication, the separation requirement between two cameras (antennas) is 500m [15, P.301] for a 6m RX antenna at the BS. In such case, a so-called *space-division multiple access* (SDMA) technique can be employed to boost the wireless link speed. In theory, under the same energy constraint, the use of SDMA doubles the maximum wireless link speed. In other words, to transmit the same number of images, a two-camera system spends half of overall communication energy as compared with a single-camera system. Therefore, despite the computation energy, the consumption per camera is reduced by a factor of 2×2. Similarly, if the number of cameras is n, then MIMO can potentially reduce the communication consumption per camera by a factor of n^2, which is a significant improvement.

VII. Conclusions

We propose a multi-camera system for environmental monitoring to increase the robustness of the system. By interleaving the sampling time of all cameras, the event detection probability is kept the same as in a single-camera system, while the energy burden is distributed among all cameras. To deal with the uncertainties in sensor network, we propose a robust adaptive synchronization scheme to build the optimal sampling structure autonomously. The theory as well as simulation results show that with help of overhearing, this algorithm converges to the optimal configuration fast, and can tolerate certain overhearing loss.

To compress the correlated videos from disjoint cameras, we propose three video coding methods, namely, distributed/independent/joint coding schemes. The evaluation results on a two-camera system shows that joint/independent video coding perform substantially better than distributed video coding. The results also suggest that in a long-range communication scenario, joint coding outperforms independent coding when the sampling rate is low.

For future work, we are planning to deploy a multi-camera system using *Sensorcam* in the Swiss Alps. An experimental deployment of a wireless sensor network of 12 solar-powered weather stations, with one autonomous wireless camera, has shown good performance and survived over an entire winter season. Finally, we showed the potential of MIMO technology for reducing communication consumption even further. We expect this technology to be employed in sensor networks in the near future.

References

[1] T. Ajdler, L. Sbaiz, and M. Vetterli. The plenacoustic function and its sampling. *IEEE Trans. Signal Process.*, 54(10):3790–3804, 2006.

[2] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange. Sensorscope: Out-of-the-box environmental monitoring. In *Proc. International Conference on Information Processing in Sensor Networks IPSN '08*, pages 332–343, 2008.

[3] Z. Chen. Sensorcam cameras and image registration. Technical Report 170457, EPFL, 2011. Available at http://infoscience.epfl.ch/record/170457.

[4] Z. Chen, G. Barrenetxea, and M. Vetterli. Distributed successive approximation coding using broadcast advantage: The two-encoder case. In *Proceedings of the 48th Annual Allerton Conference on Communication, Control and Computing*, pages 1110 –1116, Sep 29 - Oct 1 2010.

[5] Z. Chen, G. Barrenetxea, and M. Vetterli. Distributed successive refinement of multi-view images using broadcast advantage. *submitted to IEEE Trans. Image Process.*, 2011.

[6] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero. Distributed video coding. *Proc. IEEE*, 93(1):71–83, Jan. 2005.

[7] J. Hicks, J. Paek, S. Coe, R. Govindan, and D. Estrin. An easily deployable wireless imaging system. In *ImageSense'08: Proceedings of the Workshop on Applications, Systems, and Algorithms for Image Sensing*, 2008.

[8] S. Jayaweera. Virtual MIMO-based cooperative communication for energy-constrained wireless sensor networks. *IEEE Trans. Wireless Commun.*, 5(5):984–989, 2006.

[9] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. In *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, pages 96–107. ACM, 2002.

[10] H. Nguyen, B. Bhanu, A. Patel, and R. Diaz. VideoWeb: Design of a wireless camera network for real-time monitoring of activities. In *Proceedings of 3rd ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–8. IEEE, 2009.

[11] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi. Video coding with H. 264/AVC: tools, performance, and complexity. *Circuits and Systems magazine, IEEE*, 4(1):7–28, 2005.

[12] R. Puri, A. Majumdar, and K. Ramchandran. Prism: A video coding paradigm with motion estimation at the decoder. *IEEE Trans. Image Process.*, 16(10):2436–2448, 2007.

[13] M. Rahimi, R. Baer, O. Iroezi, J. Garcia, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: in situ image sensing and interpretation in wireless sensor networks. In *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 192–204. ACM, 2005.

[14] B. Song, O. Bursalioglu, A. Roy-Chowdhury, and E. Tuncel. Towards a multi-terminal video compression algorithm using epipolar geometry. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2. IEEE, 2006.

[15] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge Univ Pr, 2005.

[16] A. Wang and C. Sodini. On the energy efficiency of wireless transceivers. In *Proceedings of IEEE International Conference on Communications*, volume 8, pages 3783–3788. IEEE, 2006.

[17] D. Xie, T. Yan, D. Ganesan, and A. Hanson. Design and implementation of a dual-camera wireless sensor network for object retrieval. In *2008 International Conference on Information Processing in Sensor Networks*, pages 469–480. IEEE, 2008.

[18] C. Yeo and K. Ramchandran. Robust distributed multiview video compression for wireless camera networks. *IEEE Trans. Image Process.*, 19(4):995 – 1008, 2010.