A Color Frame Reproduction Technique for IoT-based Video Surveillance Application

Rashedul Hasan*, Shahed K. Mohammed†, Alimul Haque Khan‡ and Khan A. Wahid§
Department of Electrical and Computer Engineering, University of Saskatchewan,
Saskatoon, Saskatchewan S7N5A9, Canada
Email: *rah756@mail.usask.ca, †shahed.mohammed@usask.ca, ‡alimul.khan@usask.ca, §khan.wahid@usask.ca

***Abstract*—In this paper, we present an IoT-based power-efficient color frame transmission and generation algorithm for video surveillance application. The conventional way is to transmit all R, G and B components of all frames. Using our proposed technique, instead of sending all components, first one color frame is sent followed by a series of gray-scale frames. After a certain number of gray-scale frames, another color frame is sent followed by the same number of gray-scale frames. This process is repeated for video surveillance system. In the decoder, color information is formulated from the color frame and then used to colorize the gray-scale frames. Our experimental results show that the IoT-based approach gives better results than traditional techniques in terms of both energy efficiency and quality of the video, and therefore, can enable sensor nodes in IoT to perform more operations with energy constraints.**

Keywords*—*IoT, Video Surveillance, Color Reproduction, Energy Saving.

I. Introduction

The *Internet of things (IoT)* is the inter networking of physical devices, vehicles, buildings and other items, embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data.IoT melds together physical objects, virtual objects, living beings, user interfaces and analytics all interconnected over an internet-based infrastructure. In recent years, increasing amount of video cameras have been appearing throughout, including surveillance cameras for security, maintenance, management, healthcare and many critical purposes. *Video surveillance* sensors and surveillance data are interesting research topics in the field of IoT mainly for the three reasons. Firstly, the transmission bandwidth of a video surveillance node is much higher, leads obstacle for the communications systems. Secondly, the video surveillance cameras are usually been installed in any place including remote areas. So there are a lot of battery driven surveillance video sensor available in IoT-based application. But the power consumption of those video sensors is much higher for transmitting a huge amount of data which leads higher deployment and maintenance cost. Thirdly, the huge information gathered from distributed video surveillance nodes results in ultra-big data [1]. Therefore, video data filtering, processing and data mining is also a crucial task in IoT.

The fundamental goal of this work is to penetrate value on energy and bandwidth efficient IoT-based video surveillance system. Some prior researches have been conducted on the same objective. For video surveillance system in IoT paradigm, slot allocation of multicamera video streaming that is validated under IEEE 802.15.4e wireless personal area networks was used in [2]. In [3], a list of possible recommendations for low computation and memory capacity of a video surveillance node, connected via low-quality wireless channels was discussed. In [4], a power-scalable video encoding method was proposed to minimize the energy consumption in portable video communication devices.

In conventional video surveillance system, color images are captured by the camera sensor and then transmitted out through a wireless channel to a server where the images are stored. Transmitting all three color components (i.e. red, green and blue) of a color image consumes significant processing time and battery power for a battery driven surveillance node. In this paper, our idea is to transmit some gray-scale images from the transmitter and regenerate the color images in the receiver side. There are many color generation algorithms found in the literature that mainly provide color tone from a color image to a gray-scale image [5]. In [6], the color information is assigned to each pixel in the gray image by matching each gray-scale image pixel to a pixel in the target swatch using the Euclidean distance matric. In [7], the authors have used a set of seed points and their respective color vectors in the RGB format with a YUV-based classification. In [8], a quadratic objective function-based optimization method is used to interpolate the U and V components of the YUV color space over the entire image using a set of color scribble lines. In [9], pseudo colors are employed to colorise the gray-scale image using different 64×3 color matrices. In [10], an adaptive dictionary-based color generation scheme that is targeted for capsule endoscopy application. The original color tone can be reproduced from a theme image without introducing additional color artifact. In [11], an energy-efficient wireless video sensor node with content aware pre-processing and an energy and content aware feedback control scheme was used.

In this paper, a novel way is presented that can save a significant amount of transmission energy without the need for additional hardware resources. Instead of transmitting all color images, our proposed IoT-based approach transmit only one color image followed by a series of gray-scale images. The gray-scale images received in decoder side are then colorized by a content-aware pre-processing and *motion estimation and compensation* technique. Conventional encoders such as H.264 or MPEG are computationally complex because they use motion compensation technique to remove temporal redundancy [12]. Since the energy consumption (in the form of computational cost) of motion compensation is very high and the video surveillance node has energy constraints, we

978-1-4673-6853-7/17/$31.00 ©2017 IEEE

avoid motion compensation in video sensor node/encoder side. However, we use motion estimation technique in decoder side. The number of gray-scale images used can be controlled by the user or the system. Since no additional hardware is required in sensor node and gray-scale image sequence is mostly transmitted, a significant amount of transmission power can be saved, which results in an extended battery life of the surveillance camera.

II. System Design

The proposed technique aims to reduce IoT-based video transmission energy. Reducing the size of the transmission data with less reduction of quality of the video is the key target of this technique. The goal is to design a system which can contribute to the following video surveillance constraints: 1) As most of the IOT driven distributed video surveillance nodes are low powered and often battery driven, energy saving at those nodes are the key factor designing in IOT video surveillance application. Reducing transmission cost in encoder or transmission side can save significant energy. Our system targets to do most of the computation in decoder/receiver side. Decoders/receivers are mainly composed of distributed video servers where higher computation or higher energy is not an immense issue. 2) In video surveillance real time application, changes within consecutive two frames are very low. So temporal information can contribute significant role for computational cost reduction.

In order to achieve the goals, our plan is to scale down the size of the transmission data. To obtain this, for some images, our proposed encoder transmits the gray-scale component, not all three components. For remaining images, it transmits all the three components. In decoder, the received gray-scale images are colorized with our proposed color image reproduction technique from the color frame which was received previously.

A. Encoder Design

The encoder of video surveillance node converts the frames to YC_bC_r color space. It then transmits one color image (3 component: Y, C_b and C_r) followed by a series of consecutive gray-scale (Y) images. The concept of encoder side is shown in Fig. 1. For simplicity, let *Energy Parameter* (E_p) is considered as the number of gray-scale images used between two color images. E_p results on the bandwidth as well as required transmission power of a surveillance node. Higher E_p can save significant energy and bandwidth as it increases the number of gray-scale images while transmission. On the other hand, higher E_p has degradation effects on the quality of the reconstructed image.

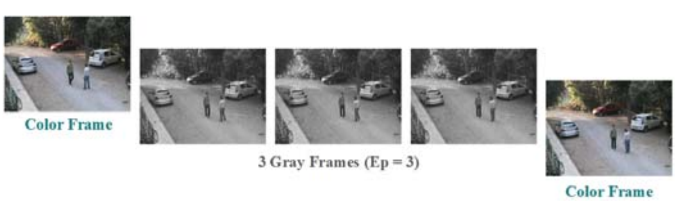


Fig. 1: Encoder sending a consecutive number of gray frames between two color frames.

Algorithm 1 Encoder Technique

1: **for** each frame **do**
2: Convert the frame to YC_bC_r.
3: Send all three components of YC_bC_r for every E_p-th frame, otherwise send only Y component.
4: **end for**

We do not let the encoder performing motion estimation in encoder. However, it is already mentioned that we use motion estimation in decoder. A color video with YC_bC_r format, Y component which represents the luminance of the color, is a good candidate of motion estimation. Moreover, C_b and C_r components are normally in half resolution of the Y component. The main advantage of selecting YC_bC_r color space for our system, is the correlation between Y and C_b, C_r which allows to reproduce high quality of the color images with only Y component. Therefore, to reduce the transmission energy and to save computational cost at encoder side, only Y component is transmitted. Algorithm 1 provides the pseudo-code for our IoT-based video surveillance encoder technique.

B. Decoder design

Decoder is designed to maximize the quality of the images with less computational cost. The algorithm and the block diagram of the proposed system are shown in algorithm 2 and fig. 2. When decoder receives an image with only Y component from the encoder, it sends the frame to a pre-processing step. In the pre-processing step, it calculates the temporal activity for each macroblock (MB) [11]. Though the standard MJPEG processes the image with the unit of 8×8 pixel block, we use different size of MB and compare the reconstructed frames. Temporal activity for each MB is measured by frame difference (*FD*) with the previous frame. After each MBs activity is calculated, MBs with activity level lower or equal than the predefined threshold are marked as *SKIP BLOCK*. Color can be directly copied from the previous frames color for a SKIP BLOCK. MBs with activity levels higher than the predefined threshold are marked as *MOTION BLOCK*. That is, MBs that have more pixels with the significant difference from the previous frame are classified as MOTION BLOCK. For each MOTION BLOCK, the motion vector is measured from the previous frame. Motion estimation and compensation are used to colorize the MOTION BLOCK. We have used three step search block matching algorithm for motion estimation.

In order to analyze moving object detection performance of the proposed preprocessing, we examine the results of frame difference operation. MOTION BLOCKs of the image without the SKIP BLOCKs are shown in Fig 3b, higher number of MOTION BLOCKs need more processing time to reconstruct color image in decoder side. However, we have measured the activity level of MBs and found that a big number of MBs having very little or no activity in the surveillance video. In Fig. 4, activity level of MOTION BLOCKs and SKIP BLOCKs are shown. If we use 15% of the maximum possible absolute difference as threshold for a video with low temporal activity, it is seen that about 98% of the MBs can be classified as SKIP BLOCK. As mentioned earlier, colors for the SKIP BLOCKs are directly copied from the previous frame's color, our technique can save significant computation energy for color

Algorithm 2 Decoder Technique

for each incoming frame **do**
  if Frame is color **then**
    Save the frame
    $Y_p \leftarrow$ Y component of the frame
  else
    for each MB of the frame **do**
      $amd \leftarrow$ absolute mean difference w.r.t Y_p for corresponding MB
      if $amd > predefined\_threshhold$ **then**
        Use block matching algorithm for motion estimation to reconstruct color frame
      else
        Copy the color of previous frame to reconstruct current frame
      end if
      Save reconstructed color frame
      $Y_p \leftarrow$ Y component of the reconstructed frame
    end for
  end if
end for

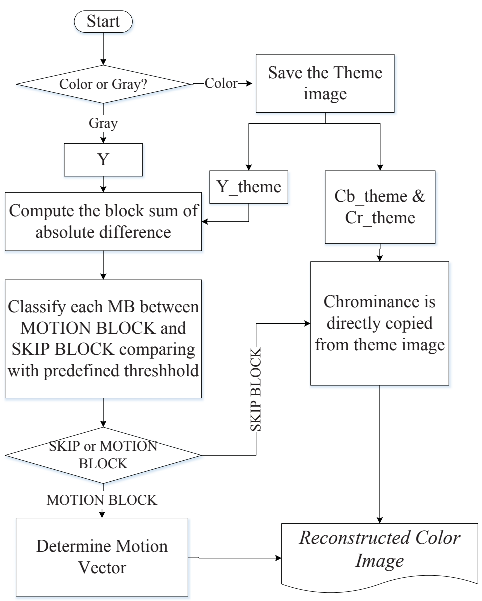


Fig. 2: Flow chart for Decoder Technique.

reconstruction.

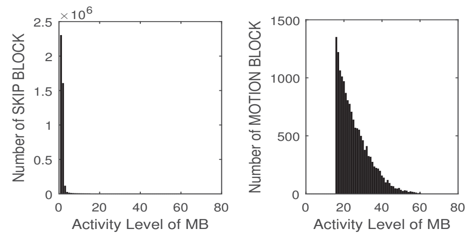
III. Results and Discussion

For performance comparison of the proposed system, we use baseline MJPEG with variable distortion and compression. The quality of the decoded video is measured by software models based on MATLAB. We use five surveillance video sequences with varying temporal activity, available at [13].

Energy consumption of the encoder side needs to keep low in order to be successful in the integration of an IoT sensor node. It is noted that the color reproduction is implemented in receiver/server and no additional hardware is needed in the

(a) (b)

Fig. 3: Result of the proposed pre-processing operation. (a) Original frame image Y component. (b) MOTION BLOCK of corresponding frame



(a) (b)

Fig. 4: Result of activity level of MB. (a) Activity level of SKIP BLOCKs (b) Activity level of MOTION BLOCKs

video surveillance node to implement it. We have computed the power consumption in terms of transmission power for the conventional (all three frames) and proposed IoT-based cases. Here, compression Ratio of the no compression, lossless compression and lossy compression are 0, 60 and 80% respectively [10]. Energy consumption per bit of the sensor node measured to be 21.05 nJ/bit [10]. Fig 5 shows the average transmission energy consumption per frame for a video of size 640×480. We can see for conventional transmission (all frames are color) the energy consumption is the highest for all cases: 165.98 mJ for uncompressed, 66.4 mJ for lossless and 33.2 for lossy images. Whereas using the proposed IoT-based technique, the energy can be significantly reduced for all cases. For example, using $E_p = 4$ (one color frame followed by four gray-scale frames), the consumption is reduced to 78, 31.2 and 15.6 mJ for uncompressed, lossless and lossy cases respectively, which results in an energy saving of 53%. In Fig. 6, estimated bandwidth savings for the different number of E_p is shown. We see that higher E_p introduces noise for reconstructed videos. But increases E_p after certain level can not improve the bandwidth savings. With this finding, we conclude that sending 4 or 5 number of gray-scale images ($E_p = 4$ or 5) between two color images is a good choice in terms of energy consumption and quality of the video.

Our scheme not only reduces significant transmission energy but also it produces a high quality of videos as we have seen it from rate-quality characteristics. We use a different number of E_p and found different compression, MSE and PSNR for each E_p. PSNR and MSE are calculated from actual color frames and reconstructed color frames. To compare, we use MJPEG as a standard for the same videos we have examined for our IoT-based technique. We vary the quality

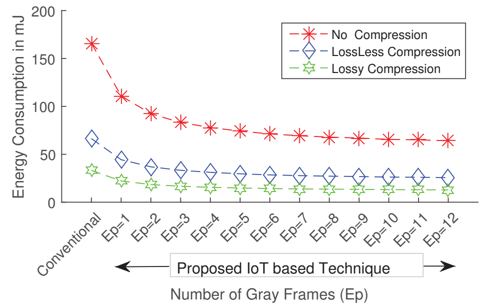


Fig. 5: Savings in transmission energy consumption for different Ep

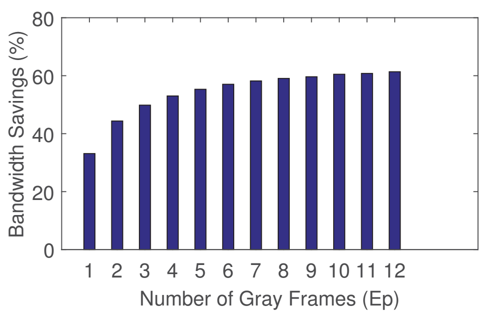
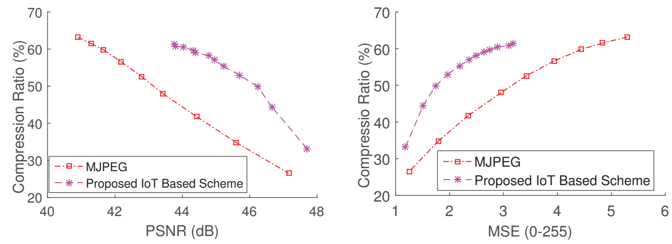


Fig. 6: Performance Comparison of required transmission bandwidth for different numbers of Ep.

factor of the MJPEG and produce multiple quality videos from the same video. We then compute the compression ratio, MSE and PSNR of those MJPEGs. In Fig. 7a and 7b, it is seen that for a fixed compression ratio, our proposed technique provides better video quality. For example, with 50% compression, we find PSNR 46 and 43.5 dB for our proposed approach and MJPEG respectively. For the same compression (50%), we find MSE 1.8 for our scheme and 2.9 for MJPEG. On the other hand, for a target PSNR of 46 dB, proposed scheme can compress 55% which is much higher than MJPEG (37%). The quality of the video in terms of MSE and PSNR of our system depends on the following: 1) Ep or number of consecutive gray frames. 2) The threshold to categorize SKIP BLOCK and MOTION BLOCK. 3) Macro block size. Changing these parameters can contribute towards the whole video surveillance quality. By implementing our technique, the video surveillance application can dynamically change E_p and threshold (for classifying MB) with the bandwidth or required quality of the video. When a quality video is important than the transmission energy savings, the system can decrease E_p and decrease the threshold to reconstruct a high-quality video. On the other hand, when the system bandwidth or channel rate becomes low or operates under a constraint energy mode, it can increase E_p to save significant bandwidth.

The most important advantage of the proposed scheme is the high efficiency in terms of energy, bandwidth and computational cost. As there is no operation introduced other than the segmenting color component in the encoder, it is of low complexity. It is also bandwidth efficient because sending single component of a color image requires less bandwidth.



(a) (b)

Fig. 7: Rate-quality characteristics of the proposed technique and MJPEG.

IV. Conclusion

We have presented energy efficient color reproduction technique that is targeted for IoT-based video surveillance application. The proposed IoT-based design achieves low-energy consumption as well as good quality reconstructed video. This scheme can enable sensor nodes in IoT to perform more operations with energy constraints. Therefore, the proposed design can be a lightweight preference to the application of video surveillance for IoT-based energy constraint sensors.

References

[1] T. Huang, "Surveillance video: the biggest big data," *Computing Now*, vol. 7, no. 2, pp. 82–91, 2014.

[2] J. Xu, Y. Andrepoulos, Y. Xiao, and M. van Der Schaar, "Non-stationary resource allocation policies for delay-constrained video streaming: Application to video over internet-of-things-enabled networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 782–794, 2014.

[3] R. Pereira and E. G. Pereira, "Video streaming considerations for internet of things," in *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*. IEEE, 2014, pp. 48–52.

[4] Z. He, W. Cheng, and X. Chen, "Energy minimization of portable video communication devices based on power-rate-distortion optimization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 5, pp. 596–608, 2008.

[5] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1120–1129, 2006.

[6] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," in *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3. ACM, 2002, pp. 277–280.

[7] T. Horiuchi and S. Hirano, "Colorization algorithm for grayscale image by propagating seed pixels," in *ICIP 2003. Proceedings.*, vol. 1. IEEE, 2003, pp. I–457.

[8] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 689–694.

[9] V. Korostyshevskiy, "Grayscale to rgb converter," *MATLAB Central file exchange [Online]*, 2006.

[10] T. Khan, R. Shrestha, M. S. Imtiaz, and K. A. Wahid, "Colour-reproduction algorithm for transmitting variable video frames and its application to capsule endoscopy," *Healthcare technology letters*, 2015.

[11] J. H. Ko, B. A. Mudassar, and S. Mukhopadhyay, "An energy-efficient wireless video sensor node for moving object surveillance," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 1, no. 1, pp. 7–18, 2015.

[12] E. Soyak, S. A. Tsaftaris, and A. K. Katsaggelos, "Low-complexity tracking-aware h. 264 video compression for transportation surveillance," *IEEE Transactions on circuits and systems for video technology*, 2011.

[13] "Video Surveillance Online Repository," http://www.openvisor.org/video_videosInCategory.asp?idcategory=20, accessed: 2016-11-03.