Compressing Surveillance Video using Motion Segmentation

Daniel Wojtaszek
SITE, University of Ottawa
dwojtasz@site.uottawa.ca

Robert Laganière
SITE, University of Ottawa
laganier@site.uottawa.ca

Abstract

In this paper, we propose a method of compressing surveillance video that, given a foreground segmentation, encodes the regions of interest using a block-based approach that reduces the distortion due to segmentation errors. Compression of each region of interest, as well as the background image, is achieved using JPEG 2000. With an estimate of the background image being updated periodically, each image is reconstructed by combining the regions of interest with the most recently encoded background image. Experiments demonstrate that using our method can result in greater compression and less distortion than other frame based compression methods.

1 Introduction

Two important aspects of networked surveillance systems that use digital imaging sensors are the storage of the image data on servers, and the transmission of the image data over the network. Storage space and network bandwidth can be very limited, therefore compression of the image data is important for the proper functioning of these systems.

For many surveillance systems the imaging sensors are stationary, so a common characteristic of the video captured using these sensors is that the amount of motion in the scene is relatively small. This results in a great amount of redundancy in the captured video. Many schemes for compressing surveillance video exploit this redundancy [3, 4, 5].

Background estimation and foreground segmentation are commonly used methods in computer assisted surveillance systems that use stationary cameras. The purpose of these methods is to determine what regions of each image in a sequence correspond to some predefined notion of activity of interest in the scene. The converse of this is that regions of the image where no activity of interest occurs can be ignored. This segmentation, therefore, can be exploited for compressing a surveillance video sequence by quantizing the background regions more coarsely than the foreground regions. Typically, the foreground regions of each image are encoded at the frame rate of the video sequence, whereas an estimate of the background is encoded at lower frame rate. Each image is reconstructed by replacing regions of the most recent background image with the corresponding foreground regions.

One common source of error in these compression methods is caused by the error in the foreground segmentation. We, therefore, propose a surveillance video compression method that reduces the effect of this error on the quality of the reconstructed video by partitioning each image into blocks, classifying each block as either foreground or background, grouping adjacent foreground blocks such that each group is rectangular, and then compressing each group as a solitary image using JPEG 2000.

JPEG 2000 is chosen as the compression standard because it has many good features for compressing surveillance video [2, 5]. Although they provide good results, compression methods that use motion compensation are considered too costly in terms of the amount computation required. The most recent standard of such compression methods is MPEG4.

This paper is organized as follows. Section 2 summarizes the previously developed methods of exploiting foreground segmentation for compressing surveillance video. In Section 3 we present our block-based method of compressing surveillance video. Finally, a comparison of the performance of our method with other methods is given in Section 4, followed by our conclusions in Section 5.

2 Related Work

The compression method proposed by Meessen et al. uses two motion JPEG (MJPEG) 2000 streams to compress a surveillance video sequence. The ROI feature of the JPEG 2000 standard is used to compress the foreground pixels to form the first MJPEG 2000 stream. The second stream contains the background images which are encoded at a lower frame rate than the original sequence. This method reconstructs each image using only foreground pixels from the current frame, replacing all background pixels in the image with pixels from the most recent background image. Therefore a foreground pixel being misclassified as background will result in an error for that pixel which may be easily visually discernible in the reconstructed image.

The method proposed in [4] partitions each frame into blocks of equal size which are each classified as either foreground or background based on a given set of criteria. Each foreground block is then encoded separately, therefore a block that is misclassified as being a part of the background will not be encoded.

3 Block-based Compression Method

This block-based compression method proceeds by computing rectangular regions of interest (ROI) using a given foreground segmentation of each image. Each ROI is then compressed as a solitary image using JPEG 2000 as is detailed in Section 3.1. Section 3.2 describes how defining each ROI by partitioning each image into blocks can reduce the foreground segmentation error. JPEG 2000 allows these blocks to be grouped into ROI. How these blocks are grouped can have a significant effect on the size of the compressed image and the quality of the reconstructed image.

Five strategies for grouping blocks into ROI are explored in this paper. For these strategies, each ROI is composed of either:

SINGLE: a single foreground block (Figure 1 (a));

FULLRECT: a rectangle of adjacent foreground blocks (Figure 1 (b));

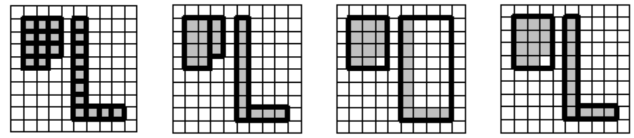
BOUNDBOX: all blocks within the bounding box of 8-connected foreground blocks (Figure 1 (c));

BOUNDFULL: blocks defined by the BOUNDBOX strategy if the number of background blocks within the bounding box is not too large, FULLRECT otherwise (Figure 1 (d)); or

FULLFRAME: the entire image.

It should be noted that the BOUNDBOX method allows background blocks to be included in a ROI. It should also be noted that there can be many different ways of computing FULLRECT. In this paper, FULLRECT first groups vertically adjacent blocks and then expands the ROI horizontally.

The BOUNDFULL strategy tries to limit the number of background blocks that are included in ROI since including too many background blocks may result in a larger bit rate with little improvement in peak signal to noise ratio (PSNR).



(a) (b) (c) (d)

Figure 1. Strategies for defining ROI given foreground blocks (gray): (a) SINGLE, (b) FULLRECT, (c) BOUNDBOX, (d) BOUNDFULL.

To determine how each of the above strategies affects image compression, a sample surveillance sequence is compressed using each strategy with $8x8$ blocks. This colour sequence shows a person walking away from the camera towards a door, and has 73 frames taken at 15 frames per second. Figure 3 (a) shows a frame taken from this sequence. Only one background image is included in the MJPEG 2000 stream for this sequence. The resulting bit rates and PSNR are plotted in Figure 2. The bit rate is the total size of the resulting MJPEG 2000 file divided by the sequence duration.

The SINGLE strategy is significantly worse than the other strategies regardless of the ROI compression quality setting, whereas the BOUNDBOX strategy achieves a higher PSNR and lower bit rate than either the SINGLE, FULLRECT, or BOUNDFULL strategies, even though this strategy resulted in more blocks being included in the ROI. This indicates that, when using JPEG 2000, it's better to combine adjacent blocks into a single ROI than to separate them. The BOUNDFULL strategy's performance is somewhere between the BOUNDBOX and FULLRECT strategies depending on the threshold used. For any given quality setting, the BOUNDBOX, BOUNDFULL, and FULLRECT strategies all result in significantly lower bit rates than the FULLFRAME strategy, although the FULLFRAME strategy results in larger PSNR values. It should be noted that a lower bit rate and higher PSNR value can be achieved using the BOUNDBOX strategy with the JPEG 2000 quality set to either 40 or 45 dB, than when using the FULLFRAME strategy with the quality set to either 30 or 35 dB.

3.1 ROI Encoding/Decoding

For a given frame, each region of interest (ROI) is compressed as a solitary image using JPEG 2000. Each frame in the Motion JPEG 2000 (MJPEG 2000) stream is composed of its set of compressed ROI. Each frame of the video sequence is reconstructed by decompressing the ROI belonging to that frame and copying them onto their corresponding regions of the current background image.

To update the background image, it is compressed us-

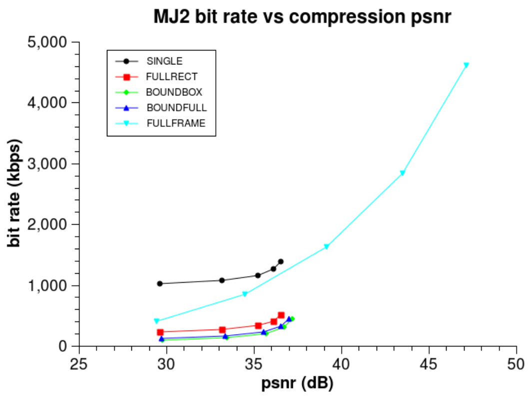


Figure 2. Plot of bit rate vs. PSNR for each ROI strategy with the following JPEG 2000 quality settings for each ROI: 30, 35, 40, 45, and 50 dB PSNR.

ing JPEG 2000 and is inserted into the current frame of the MJPEG 2000 stream as the first ROI. The background is updated once every 4 seconds, the same rate used in [5]. The only additional information required to reconstruct each frame are the number of ROI, the coordinates of the top-left corner, width, and height of each ROI, and whether the first ROI is the background image.

This method of compressing ROI requires less computation for both the encoding and decoding than the method used in [5] where the discrete wavelet transform (DWT) and the inverse DWT are computed over the entire frame. The drawback to using this method versus the method used in [5] is that compressing and decompressing each ROI separately will introduce some distortion at the edges of each ROI in the reconstructed image.

3.2 Computing ROI

Errors in foreground segmentation sometimes result in holes, outer edge deformations, and fragmentation in a foreground region, as well as small spurious foreground pixels. These segmentation errors can be reduced by partitioning each frame into blocks.

For each frame, ROI are computed by first partitioning each frame into nonoverlapping blocks that are $8x8$ pixels in size. Each block is then classified as foreground if the number of foreground pixels within that block exceeds a threshold of 3 pixels. The purpose of partitioning each frame this way is to avoid encoding any small spurious foreground pixels, to remove small holes and outer edge deformations in the foreground region, as well as coalescing any fragmented foreground regions. The ROI of a frame are composed using the BOUNDBOX strategy.

Figure 3 shows how this method of computing ROI can handle errors in the foreground segmentation of a sample frame. Figure 3 (b) shows that the foreground is very fragmented in some areas. Figure 3 (c) shows that the partitioning method coalesces most of the foreground region but some errors still remain. Figure 3 (d) shows that taking the bounding box of the foreground blocks removes most of the remaining error in the foreground region.

4 Results

To demonstrate its effectiveness, the block-based compression method described in the previous section was tested on two surveillance image sequences: the first sequence is used to show how much visible distortion is introduced by the method, the second sequence is used to compare the compression quality and bit-rate with the method proposed in [5] as well as some standard compression methods.

Figure 4 shows frames of a surveillance video in a residential neighborhood. The sequence has been captured using an Axis IP camera and transmitted to the video analytics server using MJPEG. This sequence is 212 frames long at 15 frames per second. Using the background segmentation results, this sequence is recompressed using our block-based compression scheme. It is compressed to 448 kbps with a resulting peak signal to noise ration of 33 dB. Visually, the distortions introduced by our compression method for this sequence are not obtrusive. An example of distortion caused by a block being misclassified as background is shown in Figure 5. In order for this distortion to be easily seen, we had to enlarge a portion of a frame and show the difference image between the received and reconstructed images.

To quantitatively compare the effectiveness of the block-based compression method versus other methods, it was tested using the speedway video sequence used in [5]. The original sequence, its corresponding foreground segmented binary images, and the estimated background images are available from [1].

Table 1 shows the effectiveness of various compression methods on the speedway sequence in terms of data rate and peak signal to noise ratio (psnr). Each compression method significantly reduces the data rate of the uncompressed sequence. The block-based method achieves a data rate that is 28% less, but a slightly better psnr, than the method proposed in [5]. In this way, it also outperforms MJPEG 2000 and MJPEG. The data rate and psnr for the block-based method is comparable to those achieved using MPEG4 xvid but, as previously mentioned, the block-based method does not use motion compensation and therefore is less costly in terms of computation.

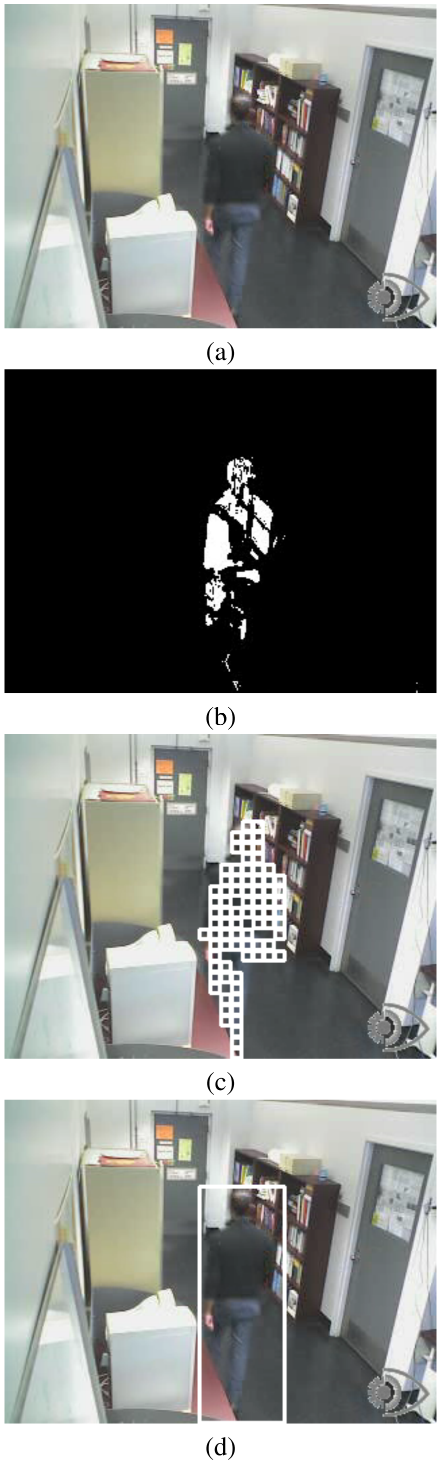


Figure 3. Example of bounding box ROI computation: (a) original frame; (b) foreground segmentation; (c) 8x8 foreground blocks with threshold = 3; (d) the computed ROI using the BOUNDBOX strategy.

Figure 4. Sample frames from a surveillance video sequence: the received image of the scene, the received foreground segmented binary image, and the corresponding resulting image after compression and reconstruction using the block-based method.

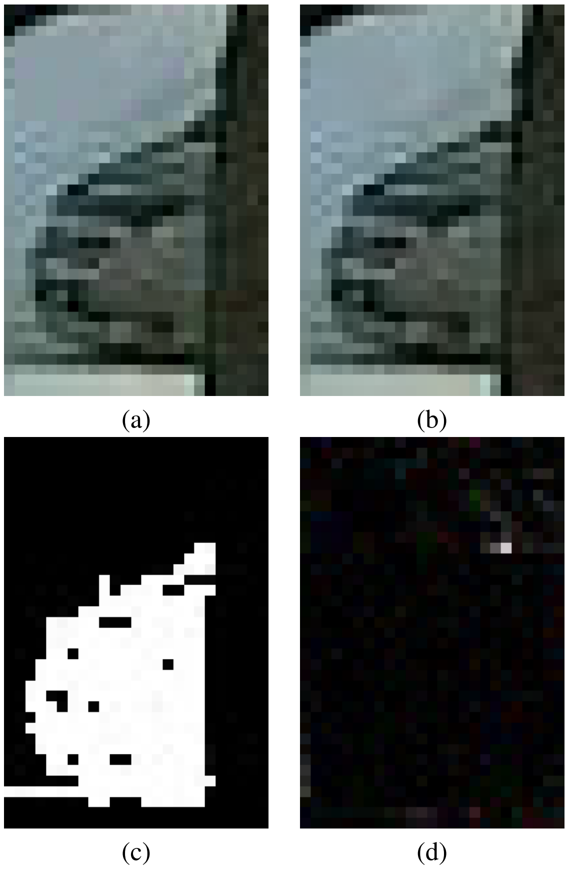


Figure 5. A zoomed in section of frame 78 from Figure 4 showing distortion in the reconstructed image due to blocks being missclassified as background: (a) the received image, (b) the reconstructed image, (c) the foreground segmented binary image, (d) the absolute pixel difference between (a) and (b).

Table 1. Data rates and psnr for various compression methods applied to the speedway video sequence. The methods are sorted in ascending order of data rate.

Method	kbps	psnr
MPEG4 xvid	458.75	37.59
Block-based	474.85	34.92
MJPEG2000 ROI [5]	660	34.7
MJPEG2000	883.91	31.43
MJPEG	1327.1	30.29
Uncompressed	30425	∞

5 Conclusions

A block-based method of compressing surveillance video has been proposed to reduce the effects of foreground segmentation errors on the quality of the reconstructed sequence. Experiments show that this method causes very little visually obtrusive distortion. A quantitative comparison of this method with standard compression methods as well as a method proposed in [5] shows that the bock-based method outperforms every method used with respect to data rate and psnr, except the MPEG4 xvid method which uses motion compensation.

References

[1] Fp6 ist-2003-507204 wcam: Wireless cameras and audio-visual seamless networking - http://www.ist-wcam.org.

[2] Iso/iec 15444-1/ iut-t t.800: Jpeg 2000 image coding system – part 1 : core coding system, 2000.

[3] R. V. Babu and A. Makur. Object-based surveillance video compression using foreground motion compensation. In *ICARCV*, Singapore, 2006.

[4] E. Magli, M. Mancin, and L. Merello. Low-complexity video compression for wireless sensor networks. In *ICME 2003 - IEEE International Conference on Multimedia and Expo*, Baltimore, USA, July 2003.

[5] J. Meessen, C. Parisot, X. Desurmont, and J.-F. Delaigle. Scene analysis for reducing motion jpeg 2000 video surveillance delivery bandwidth and complexity. In *IEEE International Conference on Image Processing (ICIP 05)*, pages 577–580, Genova, Italy, September 2005.