Video Coding Layer Optimization For Target Detection

Larry Pearlstein
Ian Patel
Department of Electrical and Computer Engineering
The College of New Jersey
Ewing, NJ USA

Alexander J. Aved
Information Directorate
Air Force Research Laboratory/RIED
Rome, NY USA

***Abstract*—One challenge in a video surveillance system is the data rate required to represent digital video. Accordingly, the use of lossy video compression at a compression ratio of 100:1, or higher, is an essential part of any distributed live video system. The ensuing distortion can interfere with the goals of surveillance by confounding both human analysis and computer vision based processing.**

This paper investigates the interaction between the video coding layer and target detection, and proposes methods for improving overall system effectiveness. Previous related research has focused on joint optimization of the video coding layer where several streams share the same bandwidth.

Our work is distinguished from prior studies in several area: we use Gradual Decoder Refresh, rather than the traditional GOP, to enable low delay and similarly avoid the use of B frames, which necessitate frame reordering. We extend the previous work by providing the ROC curves for the detection of foreground object motion, as a function of the quantization parameter. We also consider the H.265 video coding standard, in addition to H.264.

We note some surprising findings. We show that H.265 can significantly underperform H.264 in terms of Area Under Curve vs. Bitrate, and that it is possible to produce large "false alarm" blobs for moving object detection, even for a stationary, relatively noise-free source coded at low QP.

Keywords—video compression;computer vision;ROC;AUC

I. Introduction

A live-video database management system (LVDBMS) has been proposed for enhancing the effectiveness of video surveillance tasks [1]. Currently analysts can become overloaded by unimportant information. The LVDBMS enables automated event detection (AED), which can be used to provide alerts and call-outs, and thereby draw analysts' attention appropriately. AED is based on a combination of algorithms for image processing and computer vision (CV).

One challenge in a video surveillance system is the bandwidth required to store and transmit digital video data. The raw data rate for a single stream of video is typically several Gbits/sec – far too high for carriage on most wireless links and even challenging for powerful cloud computing networks. Accordingly, the use of lossy video compression at a compression ratio of 100:1, or higher, is an essential part of any distributed live video system. Although very high compression ratios are frequently necessitated by system constraints, their use can introduce significant picture distortion. This distortion can interfere with system goals by confounding both human analysis and the CV algorithms used for AED. It is therefore natural to investigate the interaction between the distortion introduced by lossy video compression and CV algorithms.

A previous study focused on joint optimization of the video coding layer, where several streams share the same bandwidth [2]. That work was intimately tied to the use of the traditional Group of Pictures (GOP) structure within the H.264 coding standard, and focused on the tailoring of bitrate and forward error correction for the specific case of detection of human subjects. Based on their framework, that work concluded that more bits should be allocated to streams where more humans had recently been detected.

For this study, we chose to focus attention on the CV operation known as "foreground detection", which can be an effective detector for object motion. Foreground detection is generally accomplished via background modeling. A recent survey paper studied a variety of methods for background modeling, and the Mixture of Gaussians (MOG) was identified as being in the group that exhibited the most robustness to challenging situations [5]. We chose to use the MOG approach for background modeling.

One of the important options involved in performing video compression is determining the rate and pattern of decoder refresh. Typical video compression systems employ *inter-frame prediction*, where the decoder is instructed to form a motion-compensated prediction of the current frame from previously decoded frames. Then only the difference between the current frame and the decoder's prediction are actually coded and transmitted. However, this iterative process requires frequent decoder re-initialization to enable a decoder to "enter" a running bitstream at a random point, and also to recover from a transmission error. For entertainment applications the frequent re-initialization is accomplished via a "Group of Pictures" (GOP) structure, which always begins with a frame that is coded in intra-frame-only mode, i.e. without referring to any previously decoded frames.

The traditional GOP structure is not compatible with low-delay applications, such as remotely controlled unmanned aerial vehicles (UAVs). The current study implements low-delay coding by using Gradual Decoder Refresh, rather than the traditional GOP, and avoids the use of B frames, which necessitate frame reordering.

We extend previous work by providing the Receiver Operating Characteristic (ROC) curves and Area Under Curve (AUC) metrics, for the detection of object motion, as a function of the quantization parameter. We derive AUC as a function of bitrate, and compare the newer H.265 video coding standard against the more widely deployed H.264 standard.

We created an experimental framework for generating content, performing video compression and applying computer vision algorithms. We ran a number of experiments – some of which confirmed our expectations, but others yielded surprising results. Section 0 describes our methodology. Our results are presented in Section II, and conclusions are given in Section 0.

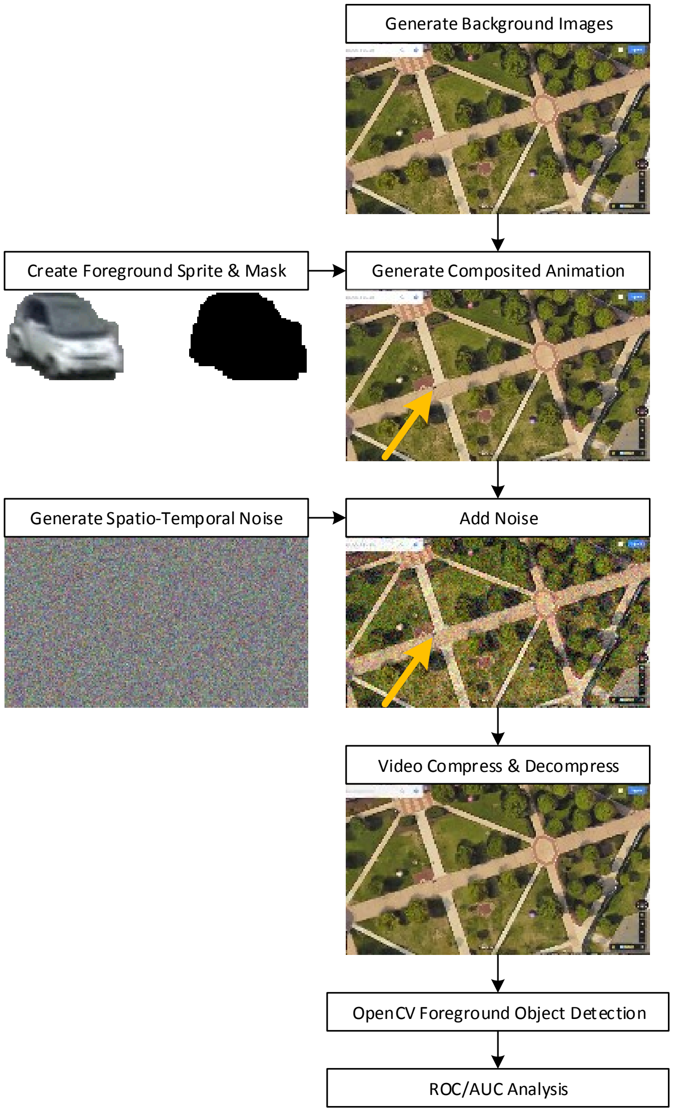
Generate Background Images
Create Foreground Sprite & Mask
Generate Composited Animation
Generate Spatio-Temporal Noise
Add Noise
Video Compress & Decompress
OpenCV Foreground Object Detection
ROC/AUC Analysis

Figure 1: Video Generation and Analysis Flow

Methodology

To explore the interaction between video compression and AED we developed an experimental framework for creating content, performing video compression applying computer vision algorithms and analyzing the results. A flow diagram that represents these steps is shown in . Each of the elements depicted is described in detail below.

A. Create Foreground Sprite & Mask

To study foreground object detection, we created a synthetic foreground image, and prepared it for animation by creating a foreground mask, as illustrated in . The foreground image sprite resembled an automobile, and had a bounding rectangle of 19Hx15V pixels, with a total of 202 active pixels under the mask. The foreground image was represented by RGB values, with 8 bits per component.

B. Generate Background Images

Three different RGB images were created for use as backgrounds, each 1920H x 1080V pixels and 8 bits per pixel component:

gray: R=G=B=128

noise: fixed high texture pattern; R, G, B, where each is composed of independent Gaussian noise, mean 128, standard deviation of 100, saturated to the range [0,255].

tcnj: Aerial view of the campus of The College of New Jersey, obtained from Google maps satellite view, which is shown in .

C. Generate Composited Animation

Object motion was simulated by overlaying the foreground sprite on top of a background image, while translating the sprite back and forth across the background image in a sequence of images.

Object-free sequences, which were used for assessing detector false alarm rates, were created by simply replicating a given background image.

D. Add Noise

Image sequences were generated by adding Gaussian noise, statistically independent both spatially and temporally, to each of the channels. The noise was scaled to produce a Peak Signal to Noise Ratio (PSNR) of either 20 dB or 40 dB, where

$$\mathrm{PSNR} = 10\,log\left[\frac{255^2}{E\left\{\left(p_{ij}-\bar{p}\right)^2\right\}}\right]$$

Here a mean value, $\bar{p}$, of 128 was used for each channel and the results were clipped to the range [0, 255].

It should be noted that a Gaussian noise PSNR of 20 dB would likely be judged as “fair to poor” quality by human viewers, whereas a PSNR of 40 dB would likely be judged as “excellent” quality [9].

E. *Video Compress & Decompress*

Video sequences were created for cases with and without an overlaid moving object, and were compressed using both the H.264 standard and the H.265 standard. The open source sequence processor "avconv" was used with the 'x264' and 'x265' CODEC libraries. Compression parameters were as follows:

Compression Attribute	**Value**
Source Image File Format	RGB, .bmp files
Encoded Pixel Format	YUV 4:2:0
# of Reference Frames	1
Refresh Strategy	GDR – vertical columns of intra coding blocks
Refresh Period	15 frames
Rate Control Method	None – constant QP
# of Bi-predictive Frames	0
Encoded File Format	.mp4, video only
Assumed Frame Rate	30 fps

F. *OpenCV Foreground Object Detection*

The processing pipeline shown in Figure 2 was implemented in C++, via the use of OpenCV 3.0 library routines, as listed in Table 1.

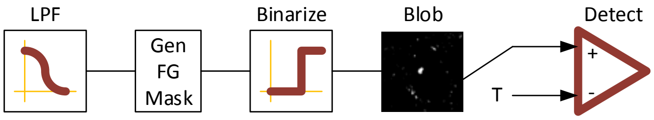

Figure 2: Foreground Object Detection Pipeline

Pipeline Function	OpenCV Library Routine
LPF	`blur()`, 3x3 kernel size
Gen FG Mask	`BackgroundSubtractorMOG2.apply()`
Binarize	`threshold()`, threshold level = 192
Blob	`connectedComponentsWithStats()`

Table 1: OpenCV Library Routines Used

The 'Blob' process created one histogram of blob sizes per frame, where we define the size of a blob as the square-root of the number of pixels in the blob:

$$blob\_histo(i,j) = \#\text{ of blobs in frame } i \text{ of size } j$$

For frame i the 'Detect' function decided for one of the following two hypotheses:

H0: No foreground object was present in the frame

H1: At least one foreground object was present in the frame

based on a threshold, T, and the detection rule:

$$\max_{j}\ [blob\_histo(i,j)] \underset{H_0}{\overset{H_1}{\gtrless}} T$$

G. *ROC/AUC Analysis*

The Receiver Operating Characteristic (ROC) is a plot of the rate of the rate of correct detection (deciding for H1 when the true state of nature is H1) vs. the rate of false alarm (deciding for H1 when the true state of nature is H0). The rates are normalized relative to unity, so take on values in the range [0.0, 1.0]. On an ROC plot the straight line through the origin with a slope of unity represents the trivial detector that decides via a biased coin flip.

We realized the state of nature, $\theta = \text{H0}$, by simply repeating the background image, adding noise and compressing. We realized the state of nature, $\theta = \text{H1}$, by compositing the foreground sprite atop the background, as described above, then adding noise and compressing. Based on varying a threshold, T, we obtained a set of operating points (correct detection vs. false alarm), and a piecewise linear curve was fit between the operating points to estimate the ROC.

The Area Under the Curve (AUC) metric refers to the area under the ROC curve, and ranges from 0.5 (for the trivial detector) to 1.0 (for the perfect detector).

II. Results

A. *Scenarios Investigated*

We investigated the impact of video compression on foreground object detection across a range of background textures, additive noise PSNRs, and QP values, for each of H.264 and H.265 compression. The full Cartesian product of parameter value ranges given in was explored. For each of the 300 cases studied, 2000 frames were encoded and decoded. The first 1000 frames were used to initialize the background model, and the ROC curves were obtained by analyzing the behavior on the last 1000 frames.

Parameter	**# of Cases**	**Range**
Synthetic Background Texture	3	{ gray, noise, tcnj }
PSNR after adding noisee	2	{ 20 dB, 40 dB }
QP values	25	{ 20, 33, … 44 }
Compression Standard	2	{ H.264, H.265 }
TOTAL	**300**	**3 x 2 x 25 x 2 = 300 scenarios**

Table 2: Ranges of Parameter Used For Investigation

B. *Connected Component Blobs*

It is instructive to examine the behavior of the system when there is no object overlaid on the background. In that scenario, we are simply compressing a still sequence with no object motion – the only variations in the source pictures are due to the

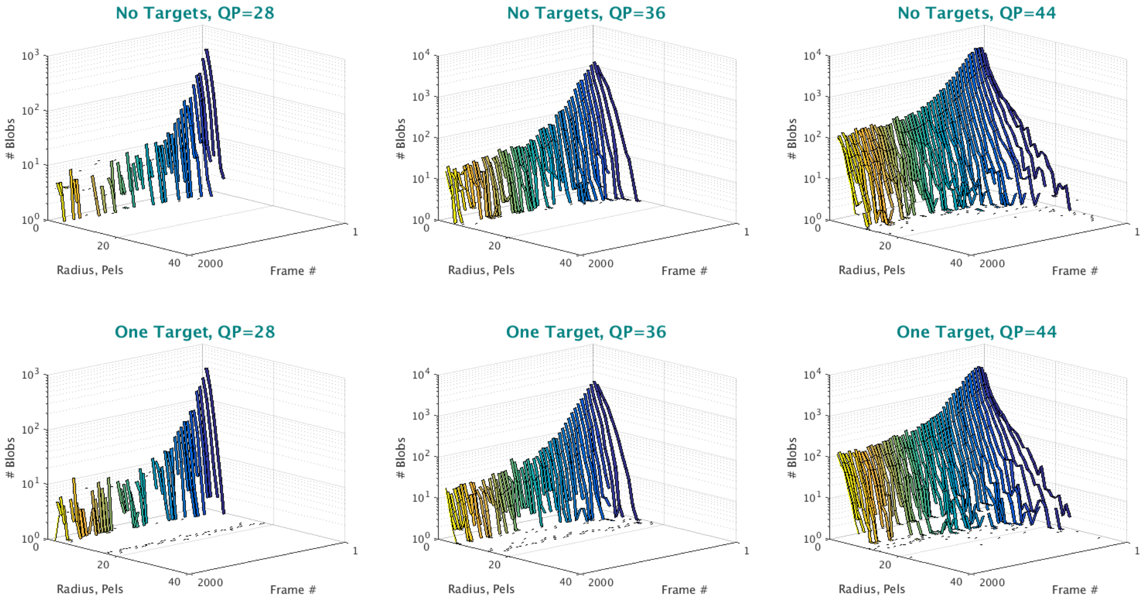

Figure 3: Blob Size Histograms for H.264 Compression, TCNJ Background, PSNR=40dB
Top Row – no foreground object present, Bottom Row – one foreground object present

additive noise, either a small amount (i.e. PSNR = 40 dB), or a large amount (i.e. PSNR = 20 dB).

1. Example

If there were no noise, the source sequence would be a perfectly static image. With a perfectly constant QP value we would expect zero frame-to-frame variations in the decoded pictures, and hence zero foreground blobs detected. Even a small amount of added noise, however, gives rise to the detection of spurious (false) foreground blobs, as shown in Figure 4. We found it somewhat surprising that large spurious blobs could be generated for this case of very low noise coded at a relatively fine quantization, at QP=34.

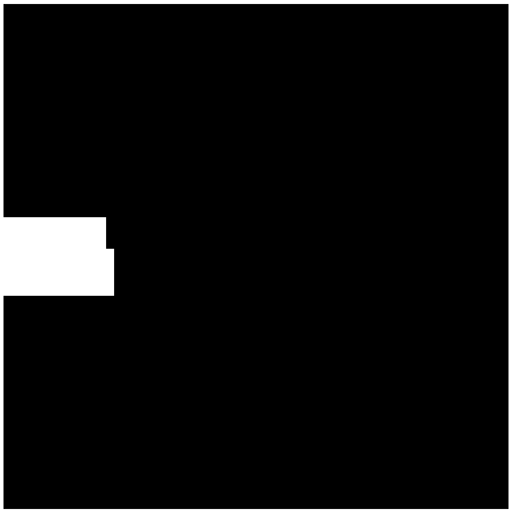

Figure 4: Spurious Blob (No Foreground Object Present) H.265 , PSNR=40 dB, Gray Background, QP=34

2. Blob Size Histograms

For illustration, blob size histograms for the ‘tcnj’ background with a PSNR of 40 dB are shown in Figure 3. The top row represents the situation where there is no foreground object present, and the bottom row represents the case of a single moving foreground object.

When the value of QP is low (left column) there is very little quantization noise introduced by video compression, and hence a relatively low rate of large spurious blobs detected. In the bottom left trace the line of detected blob sizes due to the target are easily distinguished from the much smaller blobs due to spurious detection.

When the value of QP is high there are many fairly large spurious blobs for this case, even though the source noise level is so low as to be almost imperceptible. Looking at the rightmost column the initial transient due to model initialization can be seen, lasting approximately 1000 frames.

3. Maximum Blob Sizes

It can be instructive to plot the maximum blob size per frame. This is illustrated in Figure 5, for the case of the noise pattern background, with PSNR of 40 dB. The red trace represents frames that actually had a foreground object, and the blue trace represents false blobs, obtained from frames with no foreground object. It is readily observed that there is no way to perfectly distinguish these cases for QP > 34.

C. ROC Results

An example of the ROC curves for the case of H.265 compression applied to the noise pattern background sequence with 40dB PSNR is shown in Figure 6. It is apparent that there is no ability to detect the given object on the high texture background when the value of QP is greater than, or equal to, 38.

Even for the case where QP=34 we see that perfect detection is not achieved. For example, we observed a minimum false-alarm rate of about 0.1, with no missed object detection. To put this into perspective, such a detector would issue false alarms at a rate of about 3 per second, for a frame rate of 30 fps. Clearly, such a system would be useless at relieving a human analyst from constant monitoring. A useful system might aim for a false alarm rate of one per 15 minutes, which would translate to approximately 3.7 parts per million – far too low to be accurately measured by our study of 1000 frames.

D. AUC Results

The AUCs for H.264 compression are plotted in Figure 7, and for H.265 compression in Figure 8. Given our relatively limited sample size, it is not surprising to find areas of 1.0 for relatively low values of QP. Given the relatively limited sample size considered (1000 frames) a practical system designer might reasonably aim for an AUC of 1.0. We see that, for both H.264 and H.265, an AUC of 1.0 was achieved as long as QP ≤ 32.

It would seem reasonable to expect that the AUC would decrease monotonically as QP increases, but that did not always occur. For the case of the 'gray' background at PSNR=20 dB, we observed a pronounced "valley of death", with the deepest pit at QP=40. This is due to the side-effect that video compression with extreme quantization tends to produce a noise reduction effect, by squashing high frequency DCT coefficients. At sufficiently high levels of quantization the decoded pictures tend to revert to a strongly lowpass filtered mean, and therefore exhibit very low temporal activity.

A plot of AUC vs. bitrate, which represents all of the cases examined, is shown in Figure 9. The case of the "noise" background with PSNR=20 dB fared the worst, as would be expected, based on the fact that it has the highest spatial and temporal entropy.

Perhaps the most striking result obtained is the large chasm between the performance of H.264 and H.265 on the "gray" background with PSNR=20 dB (dark red solid vs. dark red dashed). Here we found an enormous advantage for H.264. Not surprisingly H.264, with fewer coding options than H.265, does better when there is no temporal structure to the signal. This would be expected as the added modeling options cause the encoder to waste bits on symbols that indicate prediction modes. The impact is somewhat mitigated by the fact that H.265 employs context-adaptive arithmetic coding, which takes advantage of the degenerate nature of the prediction mode symbols.

An additional cause for the poor performance of H.265 in this case can be traced to the fact that, for a range of QP values, H.265 also produced worse AUC than H.264, sometimes significantly so. We observed that, for the "gray" background, the H.265 encoder produced far more pronounced artifacts due to directional prediction, than the H.264 encoder. Recall that the source image was simply flat gray, with additive noise. The H.265 encoder frequently found spurious structure in the noisy source in coding-tree units (CTUs) where intra-prediction was enforced for periodic refresh. This resulted in significant spurious detection of structured objects when those CTUs were subsequently coded without the intra-prediction constraint.

III. Conclusions

We studied the impact of video compression on the detection of foreground objects, via mixture-of-Gaussians background modeling. We found that there can be significant degradation of the computer vision algorithm due to the periodically-time varying nature of the distortion introduced by compression. As the quantizer step-size increases the detection performance generally decreases, but it can recover at very high QP values, due to the noise reduction effect that accompanies extremely high compression ratios.

We found some surprising results. We observed cases where spurious object detections can be caused even when there is almost imperceptible noise in the source sequence, even when coding at relatively low values of QP. This occurs because the background modeler "learns" that there is very low noise in the background, and sees the relatively infrequent additional distortion introduced by intra refresh as object motion. We also observed that H.265 can be far inferior to H.264 for sources with high temporal noise.

In general, we recommend the use of advanced motion-compensated noise reduction (MCNR) for surveillance systems where sensors produce a significant level of temporal noise. Many algorithms exist for MCNR – an early formulation was described in [10]. Although we found that object detection can be significantly impacted even with low source noise, the reduction of temporal noise would dramatically lower the bitrate for a given value of QP, enabling a reduction of QP, and consequent improvement in detection.

Furthermore, we are in the process of investigating an improvement to the MOG background modeler, for better dealing with compression noise. The algorithm used in this study is provided within OpenCV 3.0, and declares a background pixel based on:

```
if (totalWeight < TB && dist2 < Tb*var)
    background = true;
```

We are investigation simple modifications to that test, such as:

```
if (totalWeight < TB && dist2 < (Tb*var+qpstep2))
    background = true;
```

where `qpstep2` produces an expanded range for accepting as background a larger deviation. That increase in deviation would depend on the quantizer step size.

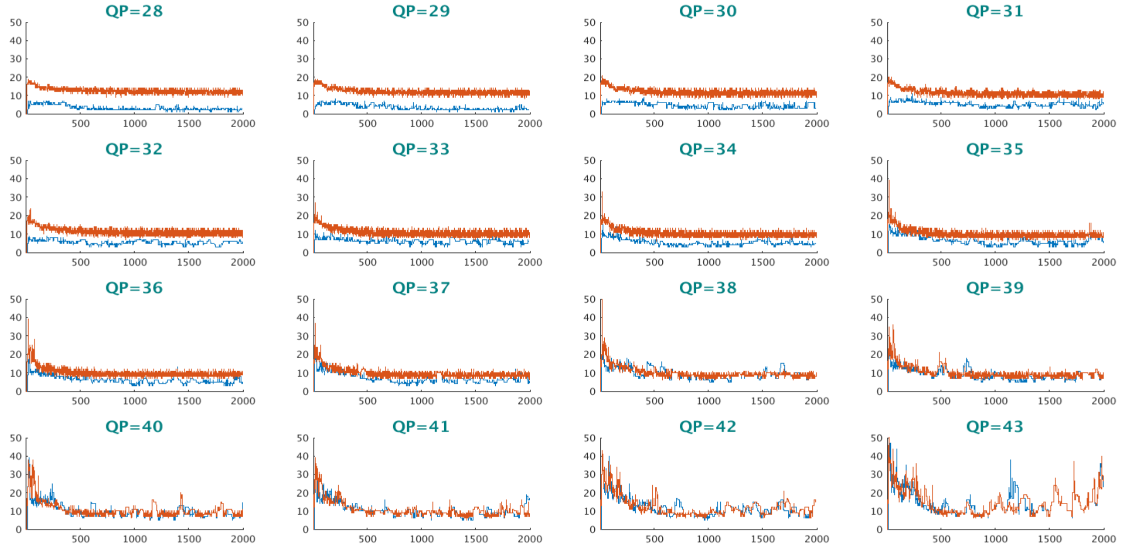

Figure 5: Maximum Blob Sizes Per Frame – H.264 Compression, Noise Background, PSNR = 40dB

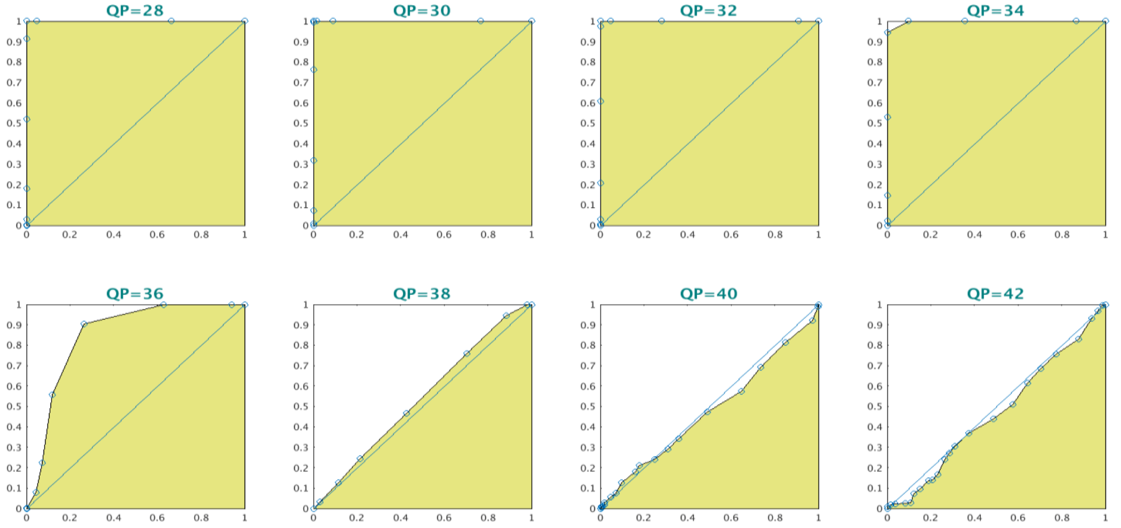

Figure 6: ROC Curves – H.265, Noise Background, PSNR=40db

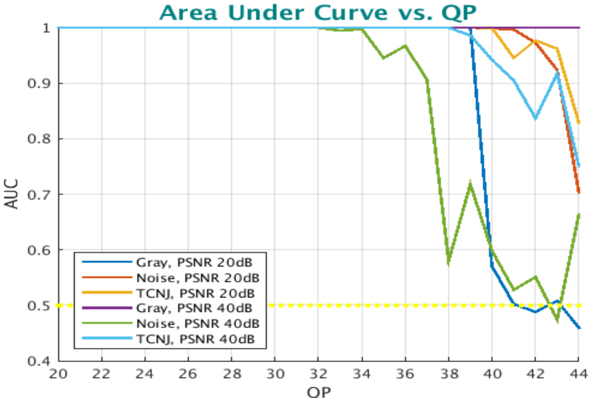

Figure 7: AUC vs. QP for H.264 Compression

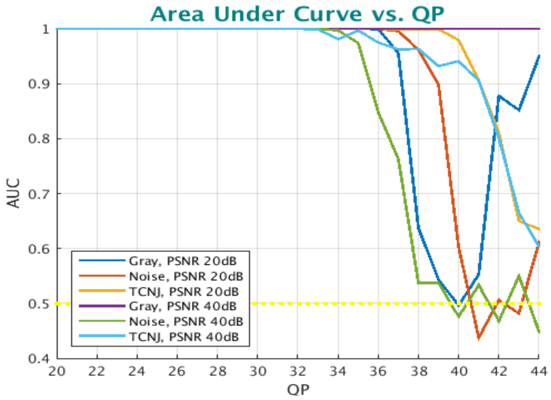

Figure 8: AUC vs. QP for H.265 Compression

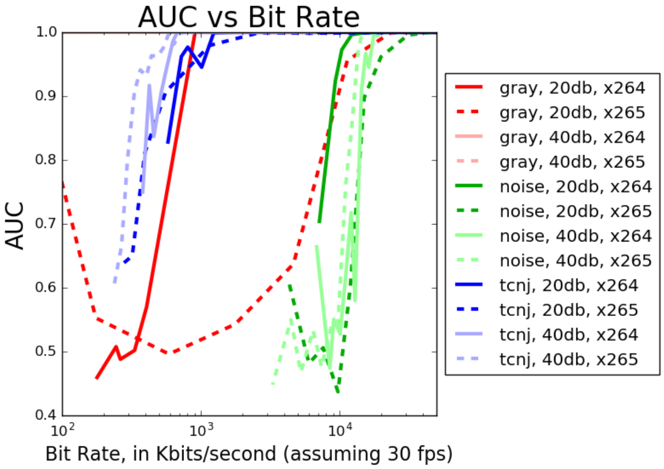

Figure 9: AUC vs. Bit Rate

REFERENCES

[1] Peng, Rui, Alex J. Aved, and Kien A. Hua. "Real-time query processing on live videos in networks of distributed cameras." Research, Practice, and Educational Advancements in Telecommunications and Networking (2012): 27.

[2] Chen, Xiang, Jenq-Neng Hwang, De Meng, Kuan-Hui Lee, Ricardo L. de Queiroz, and Fu-Ming Yeh. "A Quality-of-Content (QoC)-based Joint Source and Channel Coding for Human Detections in A Mobile Surveillance Cloud." (2016).

[3] Zivkovic, Zoran. "Improved adaptive Gaussian mixture model for background subtraction." In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 2, pp. 28-31. IEEE, 2004.

[4] Zivkovic, Zoran, and Ferdinand van der Heijden. "Efficient adaptive density estimation per image pixel for the task of background subtraction." Pattern recognition letters 27, no. 7 (2006): 773-780.

[5] Xu, Yong, Jixiang Dong, Bob Zhang, and Daoyun Xu. "Background modeling methods in video analysis: A review and comparative evaluation." CAAI Transactions on Intelligence Technology (2016).

[6] Sullivan, Gary J., Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. "Overview of the high efficiency video coding (HEVC) standard." IEEE Transactions on circuits and systems for video technology 22, no. 12 (2012): 1649-1668.

[7] Tan, Thiow Keng, Rajitha Weerakkody, Marta Mrak, Naeem Ramzan, Vittorio Baroncini, Jens-Rainer Ohm, and Gary J. Sullivan. "Video quality evaluation methodology and verification testing of HEVC compression performance." IEEE Transactions on Circuits and Systems for Video Technology 26, no. 1 (2016): 76-90.

[8] Wiegand, Thomas, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra. "Overview of the H. 264/AVC video coding standard." IEEE Transactions on circuits and systems for video technology 13, no. 7 (2003): 560-576.

[9] Sheikh, Hamid R., Muhammad F. Sabir, and Alan C. Bovik. "A statistical evaluation of recent full reference image quality assessment algorithms." IEEE Transactions on image processing 15, no. 11 (2006): 3440-3451.

[10] Boyce, J. "Noise reduction of image sequences using adaptive motion compensated frame averaging." In IEEE ICASSP, vol. 3, no. 4, pp. 461-464. 1992.