Smart Video Transcoding Solution for Surveillance Applications

White Paper

AvidBeam Technologies 12/2/15

Table of Contents

Introduction

Remote video surveillance relies on capturing video streams from camera(s) mounted on surveillance site and transmitting those streams to a remote command and control center for analysis and inspection (Figure 1).

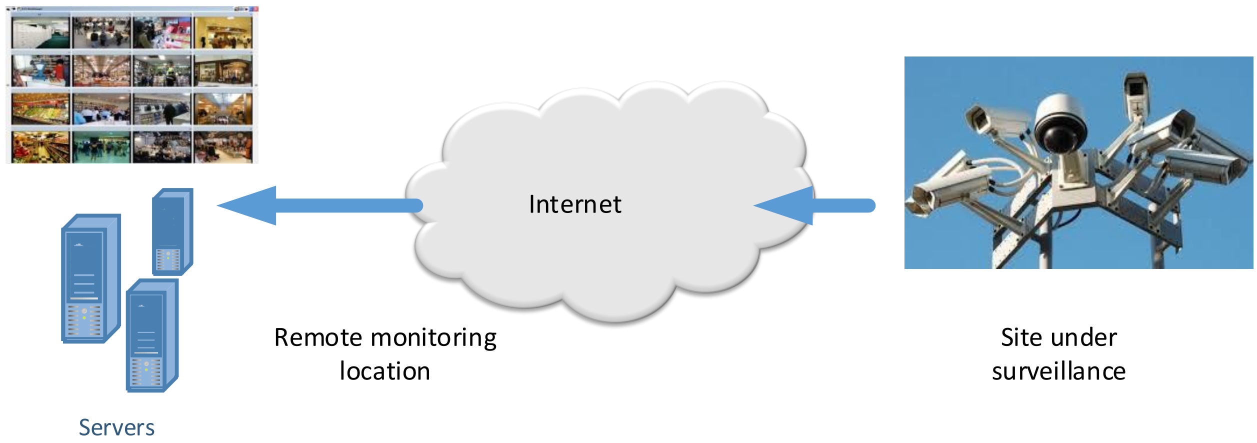


Figure 1 Remote Video Surveillance Systems

The process of streaming or uploading video to a remote monitoring location requires large bandwidth since video is streamed continuously. Moreover, many sites have large number of cameras which makes the streaming process very costly and in some cases impractical given the needed bandwidth.

In order to reduce the streaming bandwidth requirements, each camera output (frame rate, bit rate, resolution) can be configured manually which is an error prone process and would not be suitable for large scale deployment. In addition, this option can degrade the quality of important details of each video frame and make it difficult to recognize. In other solutions, video transcoding techniques (bitrate, frame rate, resolution, or combination of them)[1] are used to modify the stream to be transmitted and adapt it to the available bandwidth.

Some solutions involve the use of motion detector to identify frames with motion and transmit only those frames. This approach saves bandwidth but in many cases, especially in outdoor scenarios, it could fail since any slight motion, due to wind for example, could trigger the motion detector. The result is streaming empty frames unnecessarily.

AvidBeam Smart Video Transcoding Solution

AvidBeam has developed a comprehensive and robust solution for optimizing bandwidth for surveillance systems with limited effect to the video stream quality [2]. Our solution is based on the use of a multistage filter pipeline, as shown in Figure 2, where several filters are used to eliminate unnecessary frames and identify region of interest before invoking the video transcoder. Consequently, the transmitted bandwidth can be reduced dramatically without affecting the quality of the important information in the video frames. Clients can enable/disable each filter separately as well as configure each filter according to their needs.

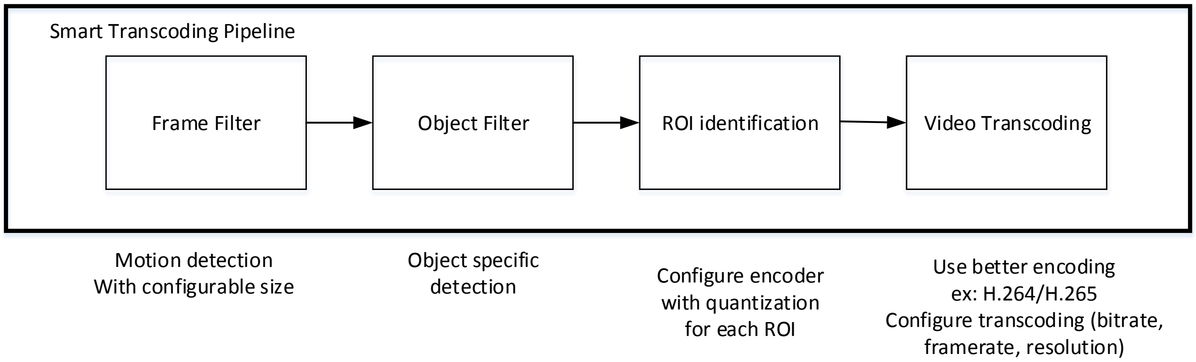


Figure 2 AvidBeam Video Transcoding Pipeline

Those filters are described as follows

1. Frame Filter:

The frame filter is used to detect motion in a given frame. The amount of motion to be detected is configured. The filter passes only frames with motion greater or equal to the configured motion size. This way, the small variation in each video frame due to external factor such as wind blow, camera vibration, or small animals or birds moving in front of the surveillance camera can be eliminated easily. Figure 3 shows the experimental results from using motion detection filter before streaming out frames. As shown in the Figure, when motion detection is enabled, frames with no significant motion are not transmitted. This results in approximately 40% saving.

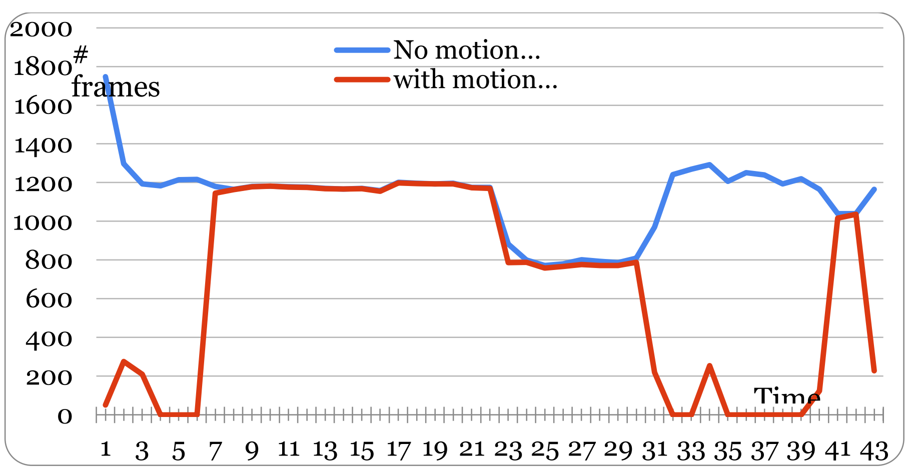
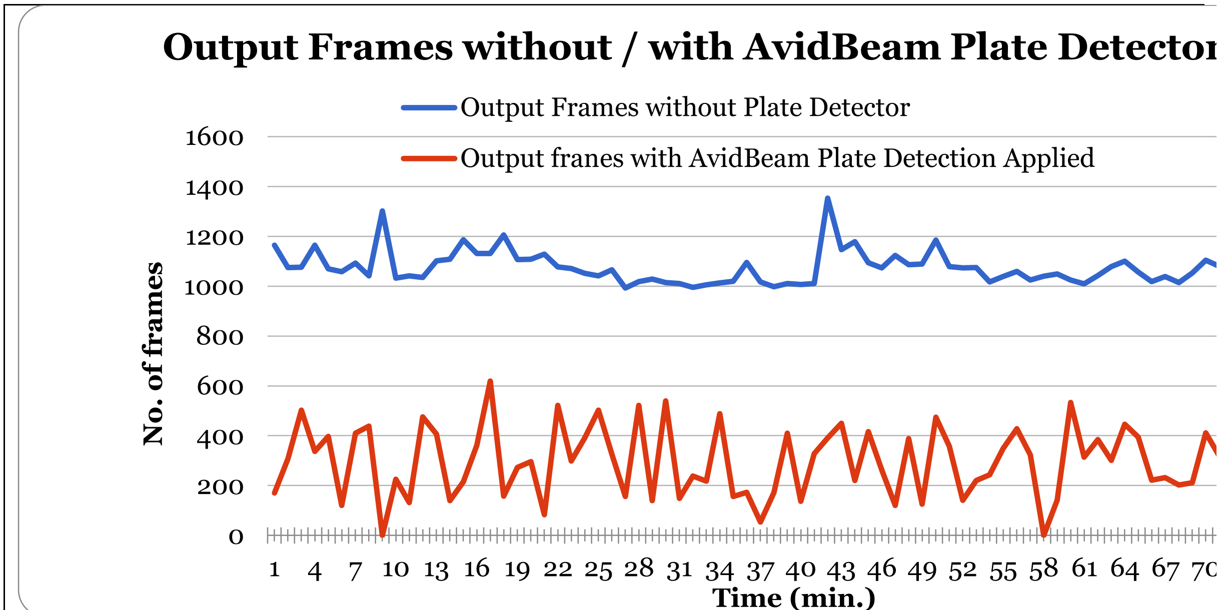


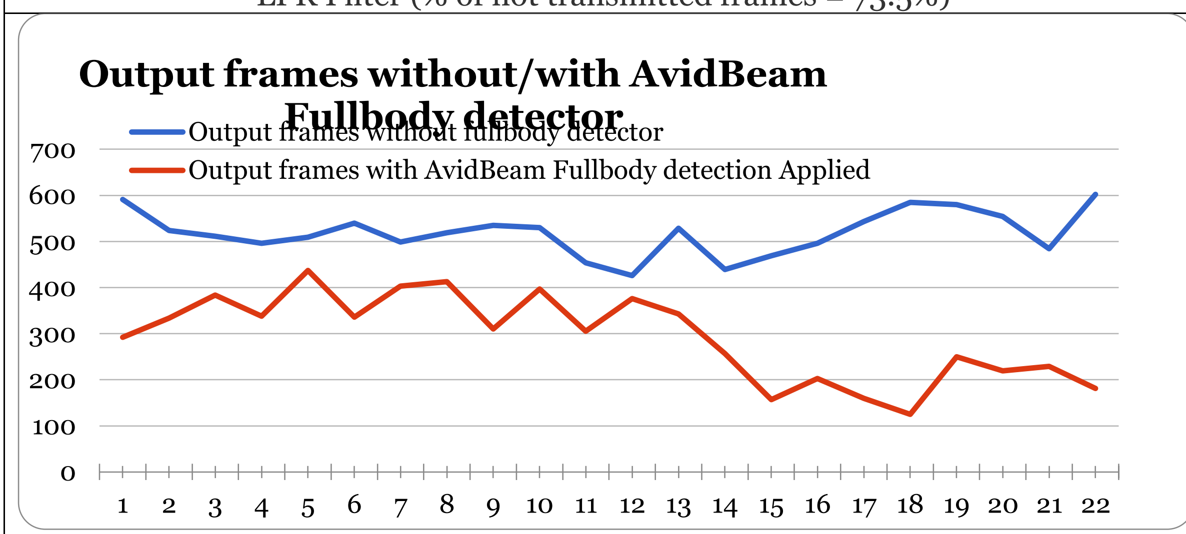
Figure 3 Number of frames with/without AvidBeam motion detection

2. Object Specific Filter:

The object specific filter is used to identify the presence of object of interest in the video frame and will pass only the frames that have the object(s) of interest. Figure 4 shows the results of applying the object specific filter to several use cases such as vehicle license plate recognition (LPR) or people count. In each case, a dedicated object detector is applied to the video frame (LPR or people). The bitrate saving results as shown in both cases which are approximately 73.5% and 43.5%. It should be noted that these results are directly proportional to the % appearance of the license plate of people in each frame.



LPR Filter (% of not transmitted frames = 73.5%)



People Body Detection Filter (% of not transmitted frames 43.5%)

Figure 4 Experimental Results for Object Based Frame Filtering

3. ROI Filter:

The purpose of the ROI filter is to identify region of interest in each frame and pass this information to the transcoder. The ROI information can be used to clip the transmitted frame or to encode the frame with different quality values for both of the ROI and none-ROI frame blocks.

As shown in Figure 5, by identifying the rectangular areas where faces are, those areas can later by encoded with higher quality compared to other areas. The impact of this approach is to obtain better quality for the ROI areas at relatively lower bitrate.

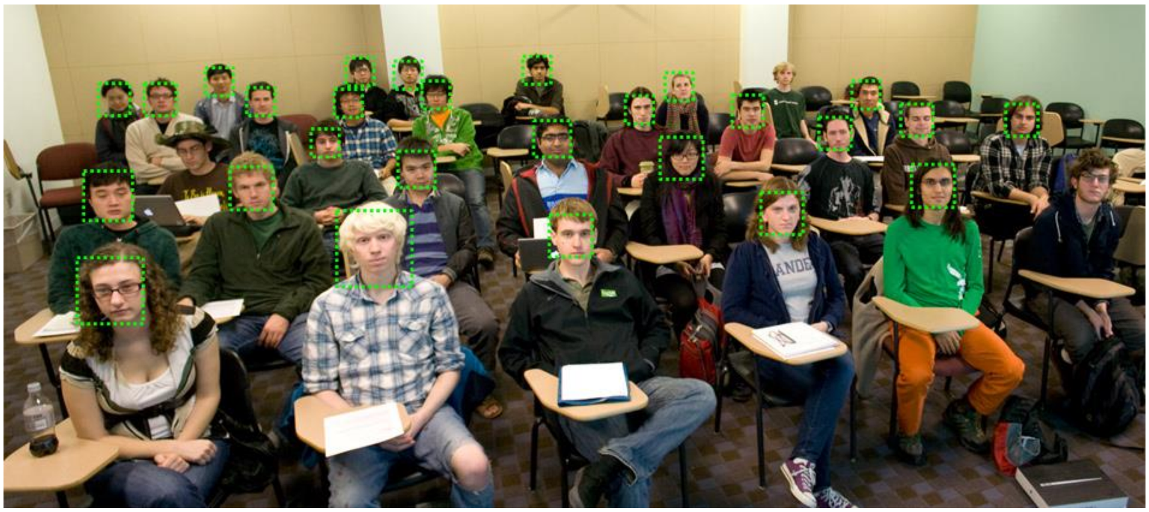


Figure 5 Face Detection – ROI

The results of applying ROI filter to the input video is shown in Figure 6. As shown compressed frame size has be reduced when ROI was enabled without affect the PSNR for the ROI areas.

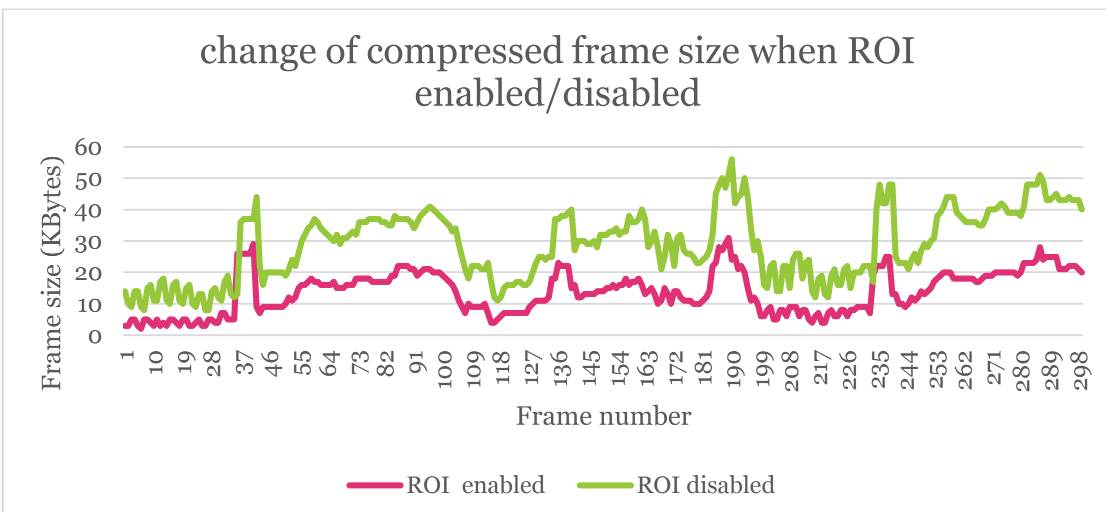


Figure 6 results

4. Video Transcoder

The final stage in the pipeline includes the actual video transcoding. The transcoder receives the selected ROI together with their proper quality (quantization) settings. Other transcoding parameters are also selected (resolution, bitrate, frame rate) based on client system configuration. As shown in Figure 7, there are several transcoding options that can be applied to the video stream.

A. **Resolution Transcoding**: in this case, each input video frame is decoded, scaled down, and re-encoded again with a new resolution
B. **Bitrate Transcoding**: in this case, each video stream is re-encoded to provide the required bitrate.
C. **Frame rate Transcoding**: in this case, the frame rate of the streamed video is modified. Frames can be dropped in order to save additional bandwidth.
D. **Video format Transcoding:** in this case, video formats that produces better bitrate for the same quality can be used. Ex H.264/H.265, MJPG/H.264, etc.)

Those options can be applied separately or combined together to achieve optimal quality/bitrate pair.

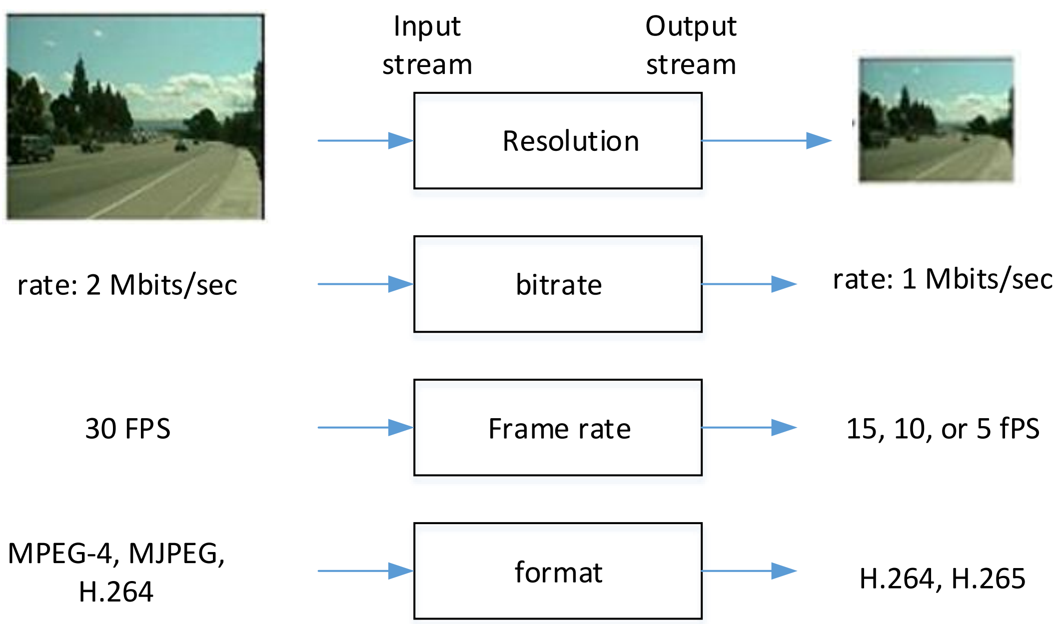


Figure 7 Video Transcoding Options

Table 1 show the experimental results for performing motion detection followed by transcoding at different bitrate (256, 512, 1024, 2048 kbps). The used video has medium amount of motion (approximately 45% of the frames have motion). Figure 8 shows the bitrate vs. time for the same results.

It can be clearly seen that using motion detection improves the bandwidth utilization by a ratio close the ratio of the frames with no motion to the total number of frames. This extra bit saving can be used to improve the video quality if necessary. With regard to quality changes, using Peak Signal to Noise Ratio (PSNR) as a quality metric, the quality produced by the two streams when motion detector was enabled/disabled were the same since the changes affected the output bitrate. There is also an applicable alternative approach in which the bitrate saving can be used to improve the quality of the video stream produced with motion detection enabled.

Table 1 Experimental Results for Combined Frame Motion Detection with Bitrate Transcoding

		Motion Detection Disabled			Motion Detection Enabled			%Saving
		Input	**Output**		**Input**	**Output**	**%diff**	
Codec		MPEG4	H264		MPEG4	H264		
Case 1 (256 Kbps)	**Measured bitrate**	2135.87	276.41	87.05	2135.87	133.38	93.76	51.75%
Case2 (512 Kbps)	**Measured bitrate**	2135.87	547.76	74.34	2135.87	262.78	87.70	52.03%
Case 3 (1000 Kbps)	**Measured bitrate**	2135.87	1068.22	49.97	2135.87	511.93	76.03	52.08%
Case 4 (2000 Kbps)	**Measured bitrate**	2135.87	2133.96	0.05	2135.87	1022.97	52.11	52.06%

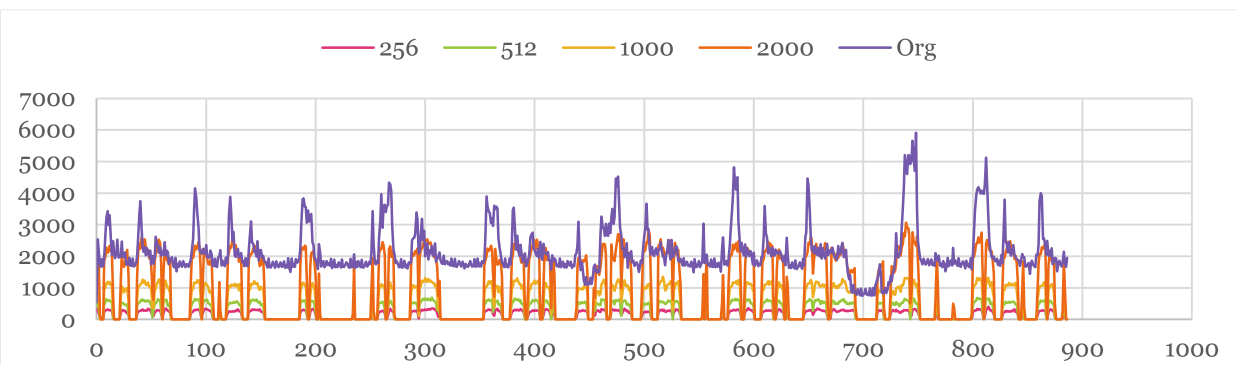


Figure 8 Bitrate v. Time for Results in Table 1

Portability Solution for Surveillance Clients

In order to avoid dependency on a particular operating system for the client of the surveillance system, there is one solution that could enable this feature. WebRTC is a browser infrastructure that enables conferencing applications. Intel developed a WebRTC SDK and is distributing to developers interested in conferencing solutions and other applications. Intel also has an open multiplexing platform for WebRTC sessions. The advantage of WebRTC is that it is embedded in the browser so it is supposedly portable across many platforms. Users would expect WebRTC enabled browsers to be available on different types of platforms and devices such as laptops, tablets and phones. This technology can enhance the transcoding solution by making it portable across platforms apart from any additional savings due to multiplexing which is also feasible. The idea is to multiplex the transcoded streams obtained from multiple cameras and transferring the multiplexed stream to the command and control center. The client at the command and control center can view the multiplexed stream on the device of their choice.

Alternatively, it is also possible to apply the transcoder after the multiplexed stream is formed. It all depends on the readiness, flexibility or limitation of the WebRTC solution. We identified a few work items that need to be completed in the solution to make it prime time ready for deployment especially in the surveillance space.

AvidBeam Transcoding Solution Components

AvidBeam Transcoding solution is used in two separate modules.

1. **AvidBeam processing unit:** This software module is installed in the surveillance site and have direct connection with the surveillance cameras. The module includes some/all of the selected filters needed to perform the necessary bitrate optimization.
2. **AvidBeam command & control center application:** This is a windows-based software application and can be installed on any device that has MS Windows. The application is used to configure the processing unit (add/remove cameras, select proper filters, as well as other transcoder configuration). The application also display the received video streams.

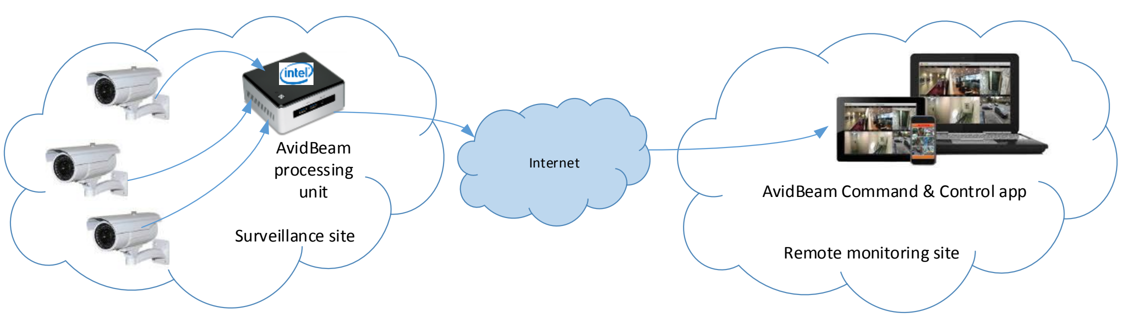


Figure 9 AvidBeam Video Surveillance Solution

The complete end-end solution is shown in Figure 9. For best performance, Intel processors are used together with its media accelerator tools. In this case, Intel GPU optimized codecs can be used to speed up the transcoding operation as well as the filter operation [3].

References

1. I. Ahmad, X. Wei, Y. Sun and Y. Zhang, "Video Transcoding: An Overview of Various Techniques and Research Issues", IEEE Transactions on multimedia, vol. 7, no. 5, Oct. 2005, pp. 793-804.
2. AvidBeam Transcoding Demo, "https://www.youtube.com/watch?v=1m_TPukXSk4"
3. Intel Media SDK white papers: https://software.intel.com/en-us/forums/intel-media-sdk/topic/311914