

PocketNetCam.com

Malaysia Leading CCTV IP Camera Solution Provider

Paul Bodell
Officer, Vice President
Sales & Marketing
IqinVision



In Part 1 of this series, we said we would discuss compression, so let's start there. While there are many different types of compression—some of the more familiar being H.264, MPEG4 and MJPEG—there are basically two types: frame-by-frame and Temporal.

A Short Note on Compression

MJPEG is the most popular frame-by-frame compression technique, compressing each image in its entirety. It is widely deployed and easy to integrate with. The advantage of this technique is that it enables you to recreate images accurately and bandwidth use is predictable. The drawback is that by handling each image in its entirety, it is not very efficient in terms of bandwidth when there is little motion or activity.

Temporal is represented by popular compression methodologies like H.263, H.264 and MPEG4. These techniques are widely deployed in applications where the available network bandwidth is limited and low quality images will suffice. Temporal takes an image, called a “key frame,” compresses it in its entirety, then for the next few images, only compresses and transmits things that change in the image. Every few images, it takes another key frame and repeats the process. The advantage of this technique is that by sending only changes to the key frame, you can save a lot of bandwidth and storage when there is very little motion or activity. One drawback of this technique is that only the key frame is a true “legal” picture. Another drawback is that motion results in a significant increase in bandwidth consumption, thereby reducing the bandwidth advantage over frame-by-frame compression.

To Megapixel or Not to Megapixel

But in reality, compression isn't significant for our following analysis, because the examples below will hold true as long as you aren't mixing apples and oranges. This article is about comparing Megapixel cameras with low-resolution cameras to see if and when deploying Megapixel makes sense. In that sense, we are simply comparing different size apples. So, for our analysis we will assume the customer wants the highest quality images available and will, therefore, use a frame-by-frame compression like MJPEG.

When does megapixel make sense? In Part 1, we talked about pixels-per-foot, which helps make the “To Megapixel or Not to Megapixel” decision fairly straightforward. Let's use the same example, but we'll now add depth of coverage as well: we want to cover a parking lot with forensic detail (40 pixels/foot) and the lot is 100 feet wide, but now let's add that we need to cover multiple rows of cars at a depth of 60 feet. To recognize faces or license plates, you would need 40 pixels/foot x 100 feet or 4,000 pixels across by 40 pixels/foot x 60 feet or 2400 pixels deep. We arrive at the total number of pixels required by multiplying pixels across by pixels deep for 9,600,000 or 9.6 Megapixel.

Megapixel Coverage vs. Conventional

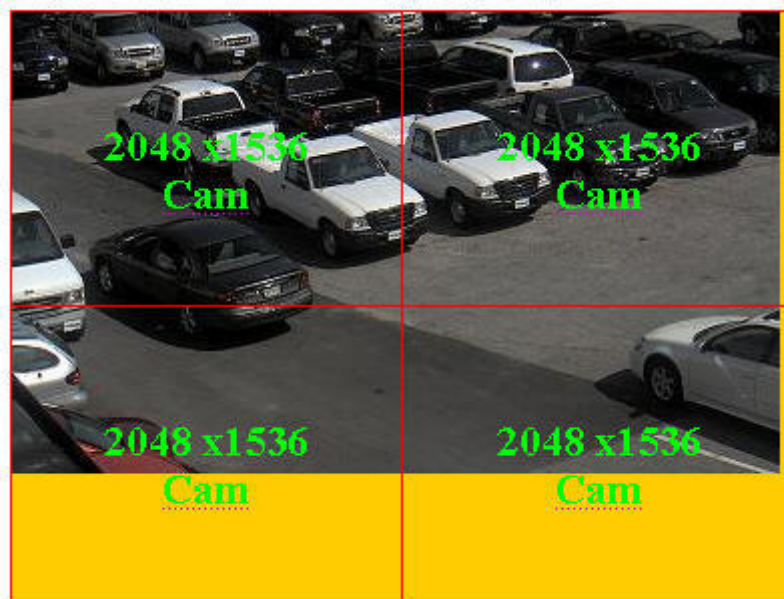
We know from our previous discussion that there are a number of cameras options to achieve forensic detail on this scene as shown in the examples below. Let's now determine the bandwidth required for each option. This is accomplished by simply calculating the size of each image and keeping the compression for all cameras the same. Once you have the image size, you multiply that by the images/second to get your bandwidth requirements. We will use a compression guideline found at a popular NVR supplier's, Milestone, website <http://www.milestonesys.com/?cid=419>. According to Milestone, a 640 x 480 image with medium-low compression has a file size of approximately 50kB. For higher resolution images we can use IqinVision's comparison matrix for megapixel cameras, (http://www.iqeye.com/iqeye/images/uploads/File/Sales_Datasheets/Product_Specs.pdf) which shows a typical 3.1 Mpix image will have a 225kB file size.

The following examples show what it would take to cover the parking lot with the desired resolution:

Option #1: 35 cameras @ 640 x 480 (4 CIF) = 35 x 50kB = 1.75MB



Option #2: 4 cameras at 2048 x 1536 (3 Mpix) = 4 x 225kB = 900kB



So, an apples-to-apples comparison reveals that Megapixel cameras can deliver the same image quality for about half the bandwidth. But let's face it; most people won't employ cameras to cover the entire area. To save money, they are more likely to only use cameras at critical "choke points," so the 35-camera-to-4-camera comparison isn't really practical. Let's take a quick look at a choke point example and see how Megapixel cameras stack up. First, let's define the choke point area; this is the area in which something you want to record is likely to happen.



Then, we run the numbers again.

Option #1: 6 cameras @ 640 x 480 (4 CIF) = 6 x 50kB = 300kB



Option #2: One camera at 2048 x 1536 (3 Mpix) = 1 x 225kB = 225kB



This example demonstrates that covering our choke point with a megapixel camera would use 33% less bandwidth than using 4CIF, 640 x 480 cameras.

And It Gets Better

Finally, some of the more sophisticated megapixel cameras incorporate smart features such as digital image cropping that allow the user to optimize a camera for a given application. Essentially they can crop out areas of the image that are not needed and save the corresponding bandwidth. In the above example, the choke point is 1920 x 960, so the 3.1 Mpix camera could be cropped to 1920 x 960 and still cover the area. This should yield a file size 135kB, which is now 42% smaller than what it would take to cover the same area with 640 x 480 cameras.

Through Parts 1 and 2 in this series, we've seen that megapixel cameras provide a compelling overall value in terms of system cost, cost-per-pixel, and overall installation cost when compared to a system comprised of fixed, low resolution cameras, but what about when you compare it to mechanical pan/tilt/zoom cameras? Stay tuned for Part 3, the final installment in our series on dispelling myths around megapixel technology.