Received: 10 November 2017 | Accepted: 2 March 2018

DOI: 10.1002/jemt.23025

RESEARCH ARTICLE

WILEY MICROSCOPY RESEARCH & TECHNIQUE

Analysis-aware microscopy video compression

Chong Shao | Jeremy Cribb | Lukas D. Osborne | E. Timothy O'Brien III | Richard Superfine | Ketan Mayer-Patel | Russell M. Taylor II

University of North Carolina at Chapel Hill, Chapel Hill, North Carolina

Correspondence
Chong Shao, University of North Carolina at Chapel Hill, Chapel Hill, NC.
Email: cshao@cs.unc.edu

Review Editor: Prof. Alberto Diaspro

Abstract

This article introduces an analysis-aware microscopy video compression method designed for microscopy videos that are consumed by analysis algorithms rather than by the human visual system. We define the quality of a microscopy video based on the level of preservation of analysis results. We evaluated our method with a bead tracking analysis program. For the same error level in the analysis result, our method can achieve 1,000× compression on certain test microscopy videos. Compared with a previous technique that yields exactly the exact same results by analysis algorithms, our method gives more flexibility for a user to control the quality. A modification to the new method also provides faster compression speed.

KEYWORDS

biomedical image processing, data compression, image analysis

1 | INTRODUCTION

The emergence of high-resolution, high-throughput microscopy systems is driving explosive growth in the size of video data produced by scientific experiments. One example of high-resolution microscopy system is described in Cribb et al. (2015). However, growth in storage capacity and data transmission is not keeping up. Consequently, video data is being compressed before transmission and storage.

Various video compression techniques have been invented and standardized in the past decades. Currently, several of the most widely used techniques are H.264, H265, and VP9 (JVT, 2003; Mukherjee et al., 2013; Sullivan, Ohm, Han, & Wiegand, 2012). Each of these video compression techniques is designed to achieve acceptable compression performance on a wide range of videos while maintaining good visual quality for human observers, sometimes based on popular video quality metrics designed to measure this such as peak signal to noise ratio (PSNR).

A method for compressing single confocal fluorescence microscopy images is presented in Amer and Dubois (2005). In their work, they estimated signal to noise ratio (SNR) in microscopy images with the techniques described in Bernas, Asem, Robinson, and Rajwa (2006) and Nowak and Baraniuk (1999). The compression was achieved by spatial downsampling, intensity downsampling, and wavelet compression.

The idea of evaluating the quality of a video based on analysis algorithms can be found in Korshunov and Ooi (2011). In their article, the video analysis routine is a set of computer vision algorithms: face recognition, face detection, and face tracking. They used H.264 to compress video multiple times with various quality settings to generate a set of the compressed video. They discovered that face recognition and face detection results are not sensitive to compression until they reach a particularly low-quality setting. Above that, compression maintains similar face recognition and detection results as the original video. Experiments have also been performed on tracking faces in a set of compressed videos with a certain portion of frames dropped. They proposed that mutual information and blackness be two computed values that better correlate to the qualities of these analysis results that they can be considered as metrics.

These metrics differ from those required by scientific analysis algorithms (Korshunov & Ooi, 2011). An analysis algorithm may require extremely high-level preservation of details in a certain region of the video, far exceeding the sensitivity of the human visual system. On the other hand, certain parts of the video that are not relevant to the analysis may be completely ignored by the algorithm but may contain noise and irrelevant moving features that greatly reduce compressibility. This mismatch has lead us to design new video compression methods for microscopy videos that we call "analysis-aware microscopy video compression."

Our efficient analysis-aware microscopy video compression method aims to provide the following features: (a) in data reduction and encoding, the method should preserve the information in a

microscopy video that is critical to analysis; (b) it should reduce information in regions not relevant to the analysis; and (c) it should be able to avoid compression and reconstruction artifacts that change the outcome of analysis even in ways that would improve results compared to the original video—analysis preformed on the compressed video should be indistinguishable from that performed on the original.

Our earlier work described a correlation-based compression method (Shao et al., 2015) that retained identical analysis results after compression. In this technique, a correlation-based threshold is used to detect the critical information (foreground) in a video. The foreground is then refined using mathematical morphology and losslessly stored in the compressed video. The remaining part is compressed by storing the temporal mean of that pixel location. We evaluated our methods using Video Spot Tracker (CISMM, 2015) to track moving beads, which showed that our method can get at most 100× compression without any change to analysis results. In comparison, H.264 compression either yielded much smaller compression ratio (lossless) or changed the analysis results (lossy).

In this article, we extend this method to enable further increase in compressibility while still maintaining results that are statistically indistinguishable from samples of the original video. We observe that microscopy video analysis results are already altered by noise introduced in all stages of the microscopy video acquisition pipeline. The new method does not force the compressed video to have identical analysis result as the original video. Instead, is maintains the original information and replicates noise such that the error introduced by compression is statistically indistinguishable from that introduced by existing noise. This is verified by run multiple different statistical analyzes on the original and compressed videos. For the case of analysis of bead-tracking results, this enables a reduction in the number of foreground pixels compared to the prior method, which enables even larger compression ratios without detectable changes in analysis.

The video compression method described in this article and in our earlier work can be characterized as region-of-interest (ROI) based methods. Previous ROI video encoding methods have been explored in Grois and Hadar (2012), Liu, Li, and Soh (2008), Van Leuven, Van Schevensteen, Dams, and Schelkens (2008). One application of ROI video coding to face detection and tracking is discussed in Menser and Brunig (2000). Application to aerial videos is introduced in Meuel, Munderloh, and Ostermann (2011). Chao et al. discussed the ROI video coding for preserving computer vision visual features in Chao and Steinbach (2011, 2012), Chao, Chen, and Steinbach (2013). To our knowledge, there is no work done in exploring the use of ROI video coding for microscopy video analysis.

Section 2 describes our analysis-aware compression technique. Section 3 explains the quality measurement being used. Section 4 describes the experiments and discusses the result. Section 5 concludes and talks about future work.

2 | MATERIALS AND METHODS

The goal of our analysis aware microscopy video compression method is to have the compressed video retain all the information required for analysis. To achieve this, the pixels that contain the useful information need to be detected in every video frame.

The basic analysis-aware compression process is illustrated in Figure 1. The basic form of our new method and our previous method both apply a two-step approach. (a) In the first (segmentation) stage, the analysis-critical regions in every frame in the video are detected. The methods both use an approach based on correlation and mathematical morphology to determine the important part of the video in a domain- and analysis-independent manner. Every pixel in every frame is labeled as either foreground or background. This result is stored in a binary map. (b) After the segmentation stage, the binary map is sent to a compression routine. The compression integrates the segmentation result in its encoding process so that for encoding setting the given fixed resource is allocated in a way to ensure that information in the analysis-critical region is well preserved. For this stage of the new algorithm, we designed and evaluated two different variations. They are detailed in Section 2.2. After the compression is completed, the resulting compressed video has a much smaller size, and it is still useful for analysis. The extended form of our new algorithm includes a third stage: (c) The compression may still introduce changes into the analysis result. To address this problem, we designed a postprocessing stage to refine the compressed video. The postprocessing stage makes use of the noise statistics in the video and refines the video by reproducing the noise that matches the video system characteristics as explained in Section 2.3.

2.1 | Segmentation stage

The goal of segmentation is to accurately detect the regions of pixels in a microscopy video frame that might affect analysis. The analysis-independent method for this task made use of the point-spread function to remove regions containing only noise as described in our previous work (Shao et al., 2015) and detailed below. That used correlation followed by the mathematical morphology "open" operation (erosion followed by dilation) to clean up small false positives and then additional dilation to expand foreground region to expand the correlation-based segmentation result. This increased dilation (shown in Figure 2) provided a conservative estimation of foreground regions to provide an (analysis-method-dependent) region increase to ensure identical results. In the new method, the expanded region is not required, so the additional dilation is not performed—resulting in a much smaller foreground region and greater compression.

The correlation-based segmentation for detecting moving objects in microscopy video is the same in both the earlier and new method. It makes use of the effect Point Spread Function (PSF) on an image. Because of the PSF, every pixel is blended slightly with its neighboring pixels. This means that any moving image feature will have a correlated impact on a region of pixels rather than only a single pixel. This does not hold for shot noise and electronic noise, which scale with image brightness but are uncorrelated between pixels. To get a foreground score for every pixel, we compute the Pearson's correlation score between it and its neighbors:

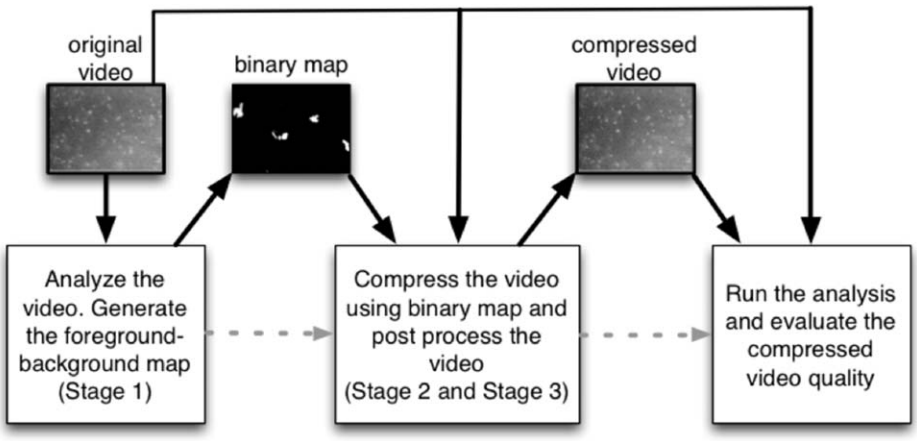


FIGURE 1 Overall analysis-aware compression process

$$R_i = \left| \frac{\sum_{j=1}^{k} (x_j - \bar{x})(y_{ij} - \bar{y}_i)}{k \sigma_x \sigma_{y_i}} \right| \quad (1)$$

In the formula, x_j is the pixel intensity value for the center pixel at the jth frame, $\bar{x}$ is the mean pixel intensity value of the center pixel over a time interval, y_{ij} is the pixel intensity value for the neighbor pixel at jth frame, and $\bar{y}_i$ is the mean for the neighbor.

We compute this value for all eight neighboring pixels for each pixel. We then compute the maximum of all neighbor scores and use a threshold on this to determine which pixels are in the foreground. The threshold was determined in our earlier study by running multiple passes of bead tracking on the compressed video has the same tracking result as the uncompressed video but it can also be determined for a system with known sensor characteristics based on a likelihood threshold based on the system's noise characteristics. Once determined, this threshold can be transferred to videos taken with similar experiment setups. After every pixel has a score assigned to it, all pixels whose score are above the threshold and are marked as potential foreground pixels in a binary map. This threshold is set to a liberal value to avoid losing actual features, with the result that the map contains many small false-positive pixel groups whose size is smaller than the PSF for a given microscope. The PSF would spread actual features over larger areas, so we remove these false positives using the mathematical erosion "open" operation. Figure 2 shows one example of the test video frame image and the result binary map cleaned up by erosion. The resulting cleaned binary map guides compression.

2.2 | Compression stage

For the compression stage, the goal is to make use of the segmentation result to encode the video data so that information in the analysis-critical regions is preserved in a manner that does not affect analysis. There are many options for applying existing well-developed video codecs and integrating the analysis-critical map signals to compress the video data. In developing our system, we explored two paths. The first approach (used both in the earlier work and the new method) processes the video frames by averaging background pixel values over time and then losslessly compressed the processed video frames using a standard algorithm. The preprocessed video has many pixel locations with constant value over time, which can be efficiently encoded to provide high compression. Tests were done to compare four standard compression techniques and software: bzip2, jpeg2000, H.264, and H.265. The result showed that the three modern compression routines all give a similar good compression with our processed video frames. From these four methods, H.265 and H.264 achieve the smallest two compressed video file size based on our data set. H.265 is 4% smaller compressed video size than H.264 but the encoding speed of H.265

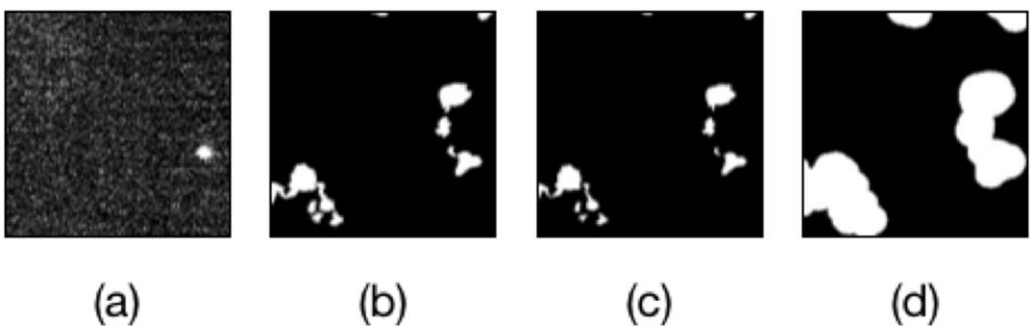


FIGURE 2 Mathematical morphology portions of the original algorithm: (a) a sub region of the first frame in the original video; (b) the correlation-based segmentation result without refinement; (c) the segmentation result after erosion refinement; (d) the segmentation result after erosion and dilation refinement. Some foreground regions in (d) are due to the other beads that move into this region in later frames

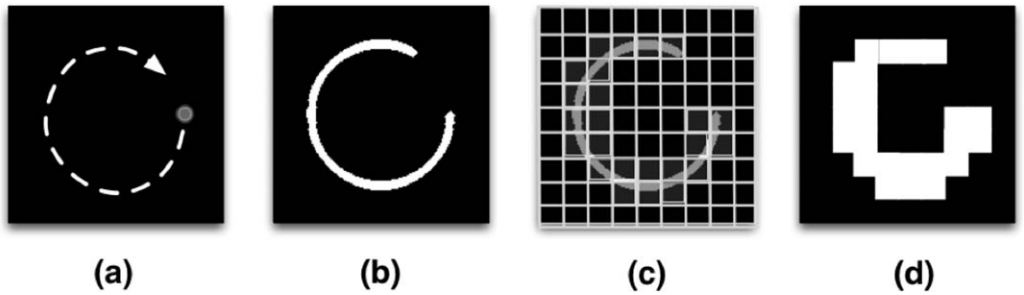


FIGURE 3 Left to right: (a) the starting position of a bead in a video and its moving trajectory; (b) the resulting binary foreground/background map; (c) illustration of the macroblocks that covers the frame; (d) the resulting binary macroblock foreground/background labeling map

was much slower than other three techniques. Therefore, we chose to use H.264 in our algorithm implementations and experiments. The results should apply to any lossless video compressor.

In an extension of the new method, we also evaluated replacing the background averaging with an approach that uses a combination of lossless and lossy compression. This approach works for block-based prediction-residual compression approaches. Our implementation used H.264. In H.264, the motion estimation unit is based on 16x16 pixel patch "macroblock." The pixel data for each macroblock is transformed into the frequency domain. Data reduction is achieved by reducing the information in the high-frequency components in every macroblock. Specifically, information reduction is done by quantization that collapses a range of close values into one. Quantization level is mostly based on the given bandwidth in compression and it is generally a global property across blocks. But in our compression, we do not need high quality for blocks that represent background pixels. Therefore, our approach assigns different quantization levels to each block based on the segmentation result. We denote qp as the controlled value in quantization. A higher qp results in a wider range of values to be suppressed into one value, which results in shorter encoded bit length and lower video quality. As shown in Figure 3, if in one block there are one or more pixels that are classified as foreground in the binary map, we use a better setting ($qp = 0$); otherwise, we assign a worse setting ($qp = 51$) to the block. This removes the need to calculate running averages across frames at the expense of variable-quantization encoding.

We also studied combining the two approaches: averaging the backgrounds and using customized qp assignment in compression. However, the combination yielded larger compressed video sizes than either technique applied by itself. This may be because the artificial edges introduced by the first stage are not usually well aligned with the macroblock boundaries, or it is not well aligned with the prediction model inside H.264.

2.3 | Postprocessing stage

By averaging the background pixels over time (V1), the compression is filtering out noise in the original video signal, producing an output video that has less noise than the input video. This modifies the results of analysis routines whose kernels reach beyond the foreground pixels, such as the symmetry-based tracking kernels uses in our analysis. This can produce more accurate tracking on the compressed video than the original. While more accurate tracking could be considered better, it is also statistically different from the results of tracking in uncompressed video. For cases where different regions of the video have different background fractions, this can also produce track-to-track variations in the results. Especially for analysis that looks at random motion distributions (like the mean-squared displacement calculations performed by our collaborators), this means that analysis on compressed video is different from analysis on uncompressed video. In these cases, the loss of noise in the reconstructed video is a problem.

There are two ways to address this problem. The original method expanding the foreground regions based on knowledge about the spatial extent of the analysis kernels (Shao et al., 2015). The new approach estimates the distribution of background noise in the original video and adds synthesized noise into the compressed video during decompression/analysis. This has the benefit of being independent of the radius of the kernels for analysis performed on the video. This process is the postprocessing stage of our method. During analysis, noise is generated and added back into the video in an online fashion. To avoid a per-pixel storage cost, the known characteristics of noise in optical microscopy systems can be used to model the entire image with only two parameters.

$$N(0,\sigma)+P(\lambda) \tag{2}$$

In estimating the noise parameters, we model noise value probability distribution as a Poisson + Gaussian distribution described in Eq. 2. By assuming a large sample size, one can further simplify the distribution (speeding reconstruction calculations) into a single Gaussian distribution with nonzero mean. The only parameters are the mean and variance of the distribution. To obtain the parameters for the two distributions (signal and noise), we used k-means clustering method. We observe that all background pixel intensity values over time in a microscopy video have similar mean values and variances. And, the foreground pixel intensity values over time tends to have diverged mean and variance statistics. Hence, we can apply *k*-means clustering method with *k* equals 2 to find the group for background pixels. By finding the two clusters of the pixel intensity over time points in the mean and variances space, we take the cluster with lower mean and variance and use its center as the mean-variance of the noise distribution. One

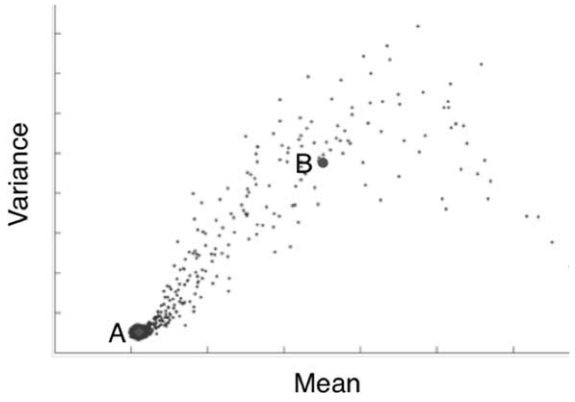


FIGURE 4 Mean vs. variation intensity plot with centers of the two-means cluster

sample plot of the pixel intensity over time's mean vs. variance plot is shown in Figure 4 where point A is the chosen cluster center.

Because standard video quality metrics such as PSNR and SSIM do not correlate well with analysis such as object tracking (Korshunov & Ooi, 2011). We seek a better metric for evaluation. In Shao et al. (2015), the quality of the video was determined by running the same tracking analysis on the video, and only the video with output exactly matching the original video's analysis result passed the validation. For our new work, we consider the fact that analysis results on the original video are affected by noise captured as part of the original video. Therefore, they represent only one of a set of possible analysis results, and re-taking new uncompressed video of the same specimen would produce slightly different results. Therefore, the compressed video's analysis does not have to exactly match that on that particular video, but it should be drawn from a distribution that matches those from multiple runs on the same specimen. We propose robust statistically-based video quality measurements based on the values derived from sets of analysis results.

This statistical approach can be used with any analysis. We demonstrate it using mean square displacement (MSD) curves that are derived from bead tracking results. An MSD value is calculated by averaging the squared displacement over movement measured using a fixed time window (τ). A sequence of MSD values with increasing time windows contains information about the type of cell motion. By characterizing at the shape of the MSD vs. τ curve, characteristics of the specimen (diffusion coefficient, membrane stiffness) can be classified (Monnier et al., 2012).

In our experiment, we converted the tracking trajectory result into a sequence of MSD values with different time windows. The quality of the video is determined based on the result of the quantitative tests on these MSD values compared to the MSD values from tracking in the original uncompressed video. We would like to verify that the error introduced by the compression matches that introduced by existing noise such that the two set of measurements are statistically indistinguishable.

2.4 | Quantitative tests

Because we judge the quality of a compressed video by comparing its error with that introduced by noise in the original video, the relative distributions should be considered. To provide confidence values, a quantitative approach in preferred. In testing the performance of our methods, we applied two experiments: the two-sample Kolmogorov-Smirnov (KS) test and Kullback-Leibler (K-L) divergence computation.

2.4.1 | KS test

In our experiment, the goal is to show that the population of MSD samples from the compressed video group is not different from the population of the samples from the uncompressed video group. This can be verified using the KS test, which is a well-known technique for testing and giving the confidence level that two groups of values drawn from two continuous random distributions are actually drawn from the same distribution. Unlike the *t*-test, which mainly tests the difference be two populations' means, the KS test takes the shape of the distribution into account and finds the largest vertical distance between two kernel density plots.

2.4.2 | K-L divergence

Computing a K-L divergence can also compare two samples of MSD values from two unknown distributions. K-L divergence is a concept in information theory that measures the difference between two probability distributions. It can be understood as the information lost when probability distributions Q is used to approximated probability distribution P. In our experiment, P is the sampled population of the MSD values for the original video and Q is the sampled population of the MSD values for a compressed video. The measurement is nonsymmetric: KL-div(P, Q) is generally different from KL-div(Q, P).

3 | RESULTS

We performed two types of experiments to evaluate our new methods. The goal is to compare the new statistically-indistinguishable analysis-aware video compression method against the standard video compression technique H.264. We included both variations of the analysis-aware video compression method (V1 from (Shao et al., 2015) and V2 from this article, using per-pixel temporal averaging) in the comparisons, and performed the comparisons for each with and without the noise-addition postprocessing. Both synthetic microscopy video data and real microscopy video data are used in the experiments.

For each compression technique, we compare the performance of different compression methods under different bandwidth settings (compression ratios). We ran the tests with various configurations to generate different compressed video sizes. We then plotted the video quality evaluation results versus video data sizes. The experiments on synthetic data and real data are discussed separately in the next two subsections.

3.1 | Experiment on synthetic data

The overall experiment flow on synthetic data is illustrated in Figure 5. We wrote a program to generate synthetic microscopy video frames. This data generating process is composed of several stages. First, we use a program to simulate bead trajectories with Gaussian random

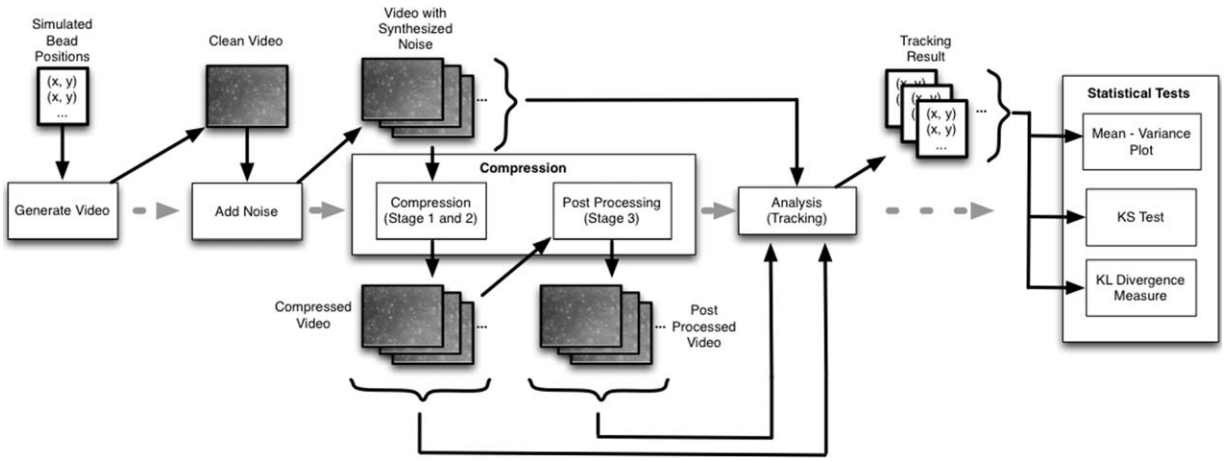


FIGURE 5 Flow chart of the experiment steps with synthetic data

walks. In this experiment, we generated 10 bead trajectories. The data was stored as a list of x-y pairs, describing the sub-pixel bead positions on every frame in the video. For an 1,800-frame video with 10 beads, we had 10 lists with length 1,800. With the bead trajectory data, we generated 10 videos that contain beads. All ten videos share the same bead trajectories.

In the second stage, for each bead position in every frame, we generate a 2D Gaussian blob with predetermined mean intensity and standard deviation values. We place it so that it is centered at the given sub-pixel *x*-*y* location based on the trajectory data list. The result is a "clean" video without background noise.

The next step is to add per-pixel noise into the video using Eq. 2. We generated the final pixel intensity values with one Gaussian plus Poisson distribution with λ equals to the pixel intensity value and σ equals to 0.01. The values are selected such at the resulting video has the similar characteristic as a real microscopy video. Therefore, in every video, the background and foreground pixels values differ, but they are samples from the same distribution.

This results in a set of 10 noisy videos. They each share the same bead trajectory, but they have different noise. Every video contains 10 beads. Every video has 1,800 frames. Figure 6a shows 1 frame in one of the 10 videos.

We tested the compression methods with the noisy videos from the data generation process. For every video, we first identify the foreground using correlation-based segmentation followed by mathematical morphology. The dilation operator size in the refinement is set to 5 pixels, which is smaller than the value we used in our previous analysis-preserving compression work. This setting does not ensure the exact same analysis results as the original video. We generate a binary map from the first step. Then we process the video with five approaches to generate 5 sets of compressed videos: (a) based on the binary map, we average the background in the video and leave the foreground unchanged. Then we compress the video using H.264, (b) Based on the binary map, we apply a customized H.264 compression by applying a low-quality setting (qp = 51) for the background pixel blocks, and applying a high quality setting (lower qp) for the foreground pixel blocks, (c) We further add synthesized noise into the resulting compressed video from (a). (d) We further add synthesized noise into the resulting compressed video from (b). To compare our method with the standard H.264 compression, in method (e) we directly compress the original video (ignoring the binary map) using H.264.

To compare the performance of the five compression methods at different quality levels, we adjust the parameters to generate a set of

FIGURE 6 Sample video frames, left: synthetic video, right: real video

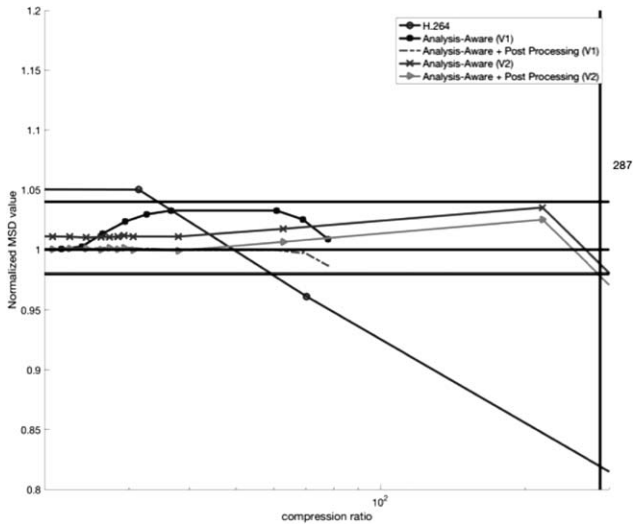


FIGURE 7 Scaled MSD values vs. compression ratio, for five groups of synthetic videos

compressed videos with different sizes for each compression method. In methods (a) and (c) we increase the dilation operator size from 5 pixels to a larger value, which produces a larger foreground region (and thus less compression). In methods (b) and (d) we used a list of qp values when compressing the foreground blocks in H.264 compression. In method (e) we used a list of qp values for H.264 compression so that we have different sized videos. We call methods (a) and (c) analysis-aware compression variation 1 (V1). We call (b) and (d) analysis-aware compression variation 2 (V2).

By including analysis-aware compression variation 1 in the experiment, we also include a generalization of the method described in (Shao et al., 2015). By increasing morphology dilation size in the refinement stage described in (Shao et al., 2015), we eventually reach a large enough dilation size that makes the analysis result the same as the one for original video.

We plot the sizes of compressed videos from various compression methods so that we can compare their relative effectiveness at a given compressed size. For each method, we ran the compression with all 10 videos to generate a population of compressed videos. The compressed video size for the videos after postprocessing is considered the same as before post processing because the post processing can be performed during reconstruction using the compressed video before postprocessing as input.

We analyzed the tracking of beads using video spot tracker (CISMM, 2015). We applied the tracking on the original uncompressed video and all compressed videos: before post processing (a and b) and after post processing (c and d). We then computed the MSD for the tracking results. Figure 7 shows the relationship between MSD values and video compression ratios. The MSD values were plotted as a multiple of the MSD values for original video (1.0 means the MSD values are identical to the ones from original videos). Each curve represents the mean MSD values among 10 copies of videos with the same foreground and different instances of sampled noise values from the same noise distribution.

The result suggests that two variations of our compression method both generate a higher quality compressed video because the points on the curves are all close to one (in the range between 0.98 and 1.04) when the compression ratio is less than 287 whereas standard H.264 yields a curve far away from 1.0 (0.82) given the same compression ratio. The curves for the videos after postprocessing are closer to the 1.0 horizontal curve, indicating that postprocessing of adding back-noise improves the video quality regarding analysis. V2 with postprocessing is very close to the original result, indicating that it may be indistinguishable from it.

3.1.1 | KS test

To further probe for potential differences between the original and compressed analysis, we performed the KS test on the MSD values. In this experiment, we used one bead's MSD values across 10 versions of the videos that share the same foreground content. After that, we

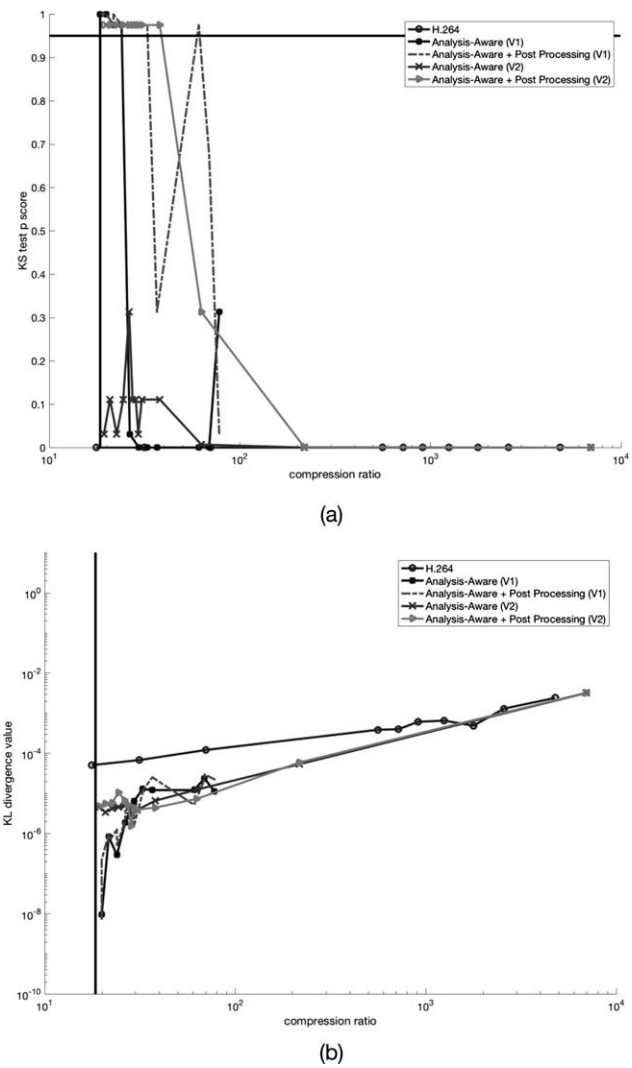


FIGURE 8 (a) KS test *p* values vs. compression ratio. The horizontal line shows the KS test *p* score 0.95. (b) K-L divergence values vs. compression ratio, for five groups of synthetic videos. The vertical line in every plot shows the compression ratio for the previous analysis-preserving method

TABLE 1 Achieved compression ratios for applying analysis-aware methods and lossless compression method on synthetic data and real data and maintain KS test p-score larger than 0.95 in the resulting video

Synthetic data (v1)	Synthetic data (v2)	Synthetic data (lossless)	Real data (v1)	Real data (v2)	Real data (lossless)
32.9	39.3	18.4	3580	23.1	9.7

For synthetic data, an improvement of around a factor of 2 was achieved above the earlier lossless method. For real data, which had more noise, the improvement was around a factor of 350.

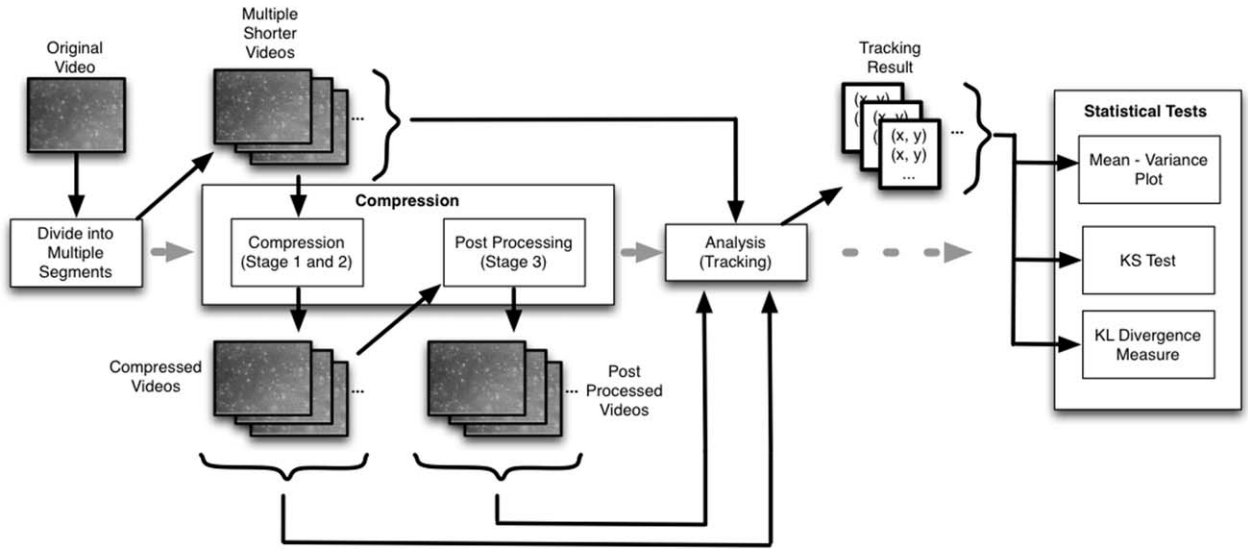


FIGURE 9 Flow chart of the experiment steps with real data

selected a fixed window size. Figure 8a shows the p values output from KS test for MSD values on videos compressed using our compression approach variation 1, our compression approach variation 2, and the standard H.264 compression. The curves plot p value vs. compression ratio. The horizontal line indicates the test decision threshold. For all p values above the line, the null hypothesis is not rejected, which means that there is no strong evidence that the MSD values obtained from compressed videos are sampled from a different population than those from the MSD values obtained from the original video (that is to say that they are statistically indistinguishable). The maximum compression ratio achieved with p values above the $p = .95$ threshold using our method is shown in Table 1.

The KS test for standard H.264 compressed videos always rejects the null hypothesis, indicating that the distributions are statistically distinguishable. For our approach before the postprocessing the curve sometimes goes above the threshold, but it also falls below the threshold as compression ratio increases. For the video compressed with our approach after postprocessing, the curves are always above the threshold until it reaches a high compression ratio of 3580.

3.1.2 | K-L divergence

We also computed K-L divergence values on the same data. The result is shown in Figure 8b. A lower K-L divergence value suggests a smaller distance between the compressed video's MSD value population and the original video MSD value population.

In this experiment, our compression approach variation 2 outperformed standard H.264 up until a compression ratio of 1450.

3.2 | Experiment on real data

We also performed experiments on videos from real experiments. For real data, it is impossible to get the true bead trajectory and generate

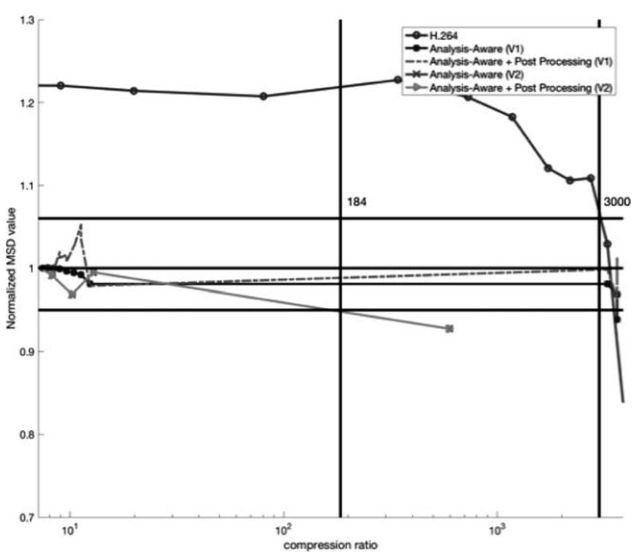


FIGURE 10 Scaled MSD values vs. compression ratio, for five groups of real videos

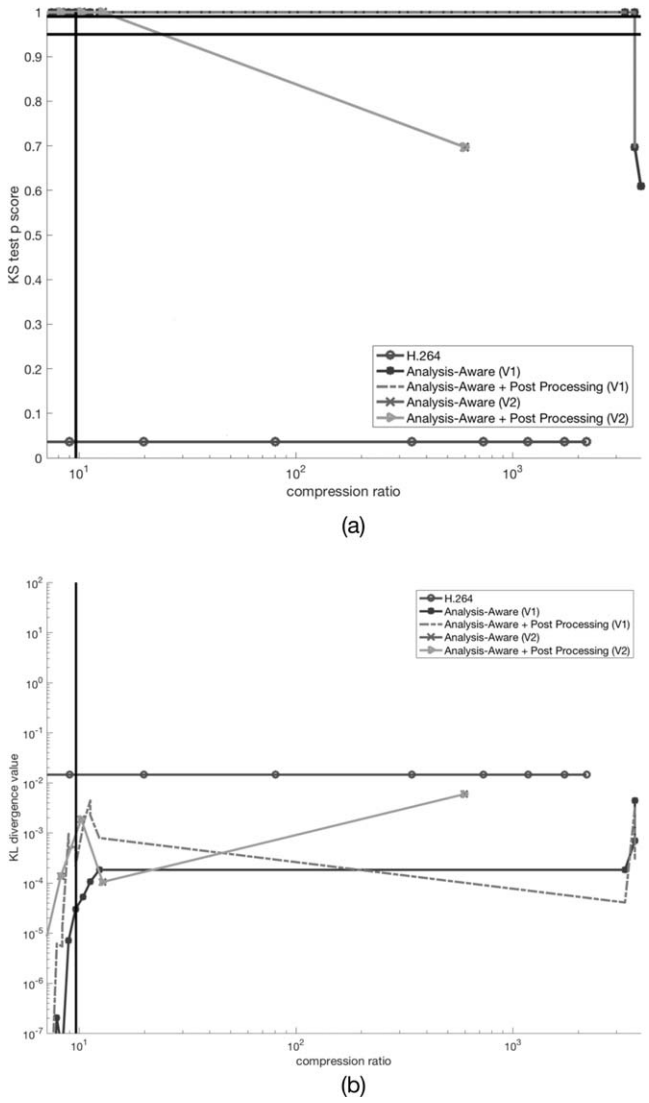


FIGURE 11 (a) KS test p values vs. compression ratio. The two horizontal lines showing the KS test p score 0.95 and 0.99, respectively. (b) K-L divergence values vs. compression ratio, for five groups of real videos. The vertical line in every plot shows the compression ratio for the previous analysis-preserving method

multiple copies of the same bead trajectory with different background noise. We handled this by dividing each video into 10 segments and performing tracking on each segment to produce a population of estimates. It is assumed that there is no significant background noise property change across the videos in a single video from the test set. The experiment process is illustrated in Figure 9.

The scaled MSD values for various compression methods vs. video compression ratio plot for real video data is given in Figure 10. For real data, a change in dilation size in our method V1 makes the compression ratio jump from 12 to ~3,000. It does not produce compressed video with compression ratio between this range. The plot shows that our compression approach also performs better than H.264 compression for compression ratios less than 3,000×. Our method generates result values near 1 (in the range between 0.95 and 1.06) for compression ratio less than 184 whereas standard H.264 compression generates values larger than 1.21 for the same compression ratio.

Figure 11 shows the MSD values from multiple compression methods compared using the KS test p-value and the K-L divergence value. For this data set, K-L divergence shows that all variations of our method outperform standard H.264. In the KS test, the p values for the videos compressed by standard H.264 are always below the $p = .95$ threshold, while our compression method variation 2 remains above the $p = .99$ threshold through very large compression ratios (>3,000×).

3.3 | Discussion: Other statistical tests

In additional to KS test and K-L divergence, we conducted further experiments with five other statistical test methods: mean test, variance F-test, Van der Waerden test, Two One-Side Test (TOST) and B-test. B-test is a maximum mean discrepancy kernel two-sample test. The goal of this experiment is to further examine the idea of evaluating video quality with statistical tests: a video with good quality under evaluations with KS test and K-L divergence is expected to also give consistent outcomes under other types of statistical tests.

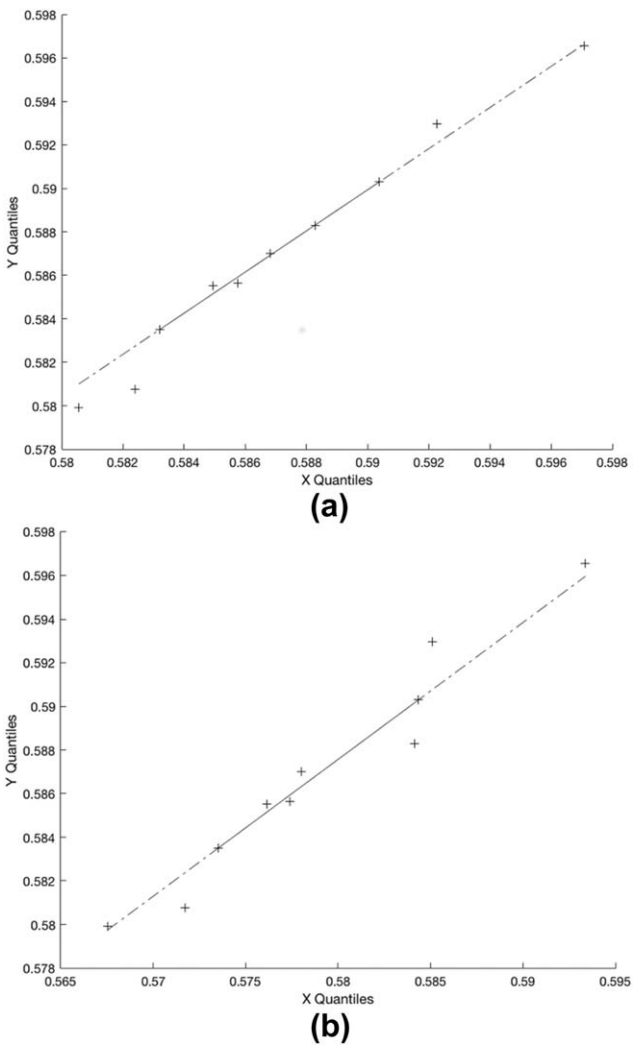


FIGURE 12 Q-Q plots. (a): less compressed data vs original data; (b): more compressed data vs original data

TABLE 2 *p*-values from different tests

Method	More compressed data vs original	Less compressed data vs original
KS test	0.031	1
Mean test	0.0145	0.9563
Variance F-test	0.2733	0.8998
Van der Waerden test	0.0181	0.8821
TOST	p1 = 0.2423, p2 = 4.3787e-06	p1 = 1.6223e-04, p2 = 2.0743e-04
B-test	0.0432	0.5097

In all tests, we used three data vectors. Each vector contains 10 numbers. The first data vector v1 is the average MSD analysis result from 10 compressed videos using analysis-aware compression method V2 with post processing, in a low video quality setting. The second data vector v2 is from 10 compressed videos using analysis-aware compression method V2 with post processing, in a high video quality setting. The third data vector v0 is from the 10 videos before compression.

First, we compared v0 with v1, v0 with v2 using two Q-Q plots shown in Figure 12. By comparing the Q-Q plot on v0 and v1 (Figure 12b) and the Q-Q plot on v0 and v2 (Figure 12a), we can conclude that a compressed video with a higher quality setting has the property closer to the original video.

Second, we apply mean test, variance F-test, Van der Waerden test, TOST and B-test on the test data. We ran 5 tests between [v0, v1] (Table 2, Column 2 from left) and [v0, v2] (Table 2, Column 3 from left). The resulting p values are listed in Table 2. In the experiment, TOST is set with lower limit −0.01, higher limit 0.01, column B is between v1 and v0, column C is between v2 and v0. We also include the result from KS test.

The table suggests that a compression with a high-quality setting (v2 vs v0) results in a higher *p* value, except the p1 value from TOST. In KS test, mean test and B-test, the *p* value of testing v1 and v0 is less than 0.05 (0.031, 0.0145, 0.0432). While *p* value of testing v2 and v0 is much higher (1, 0.95, and 0.5).

As a conclusion, our idea of measuring video quality is very generic since it can be applied with a wide variety of 2-sample statistical tests with consistent results.

4 | CONCLUSION

We introduce a compression method that preserves analysis-critical information in real microscopy video even at extremely high compression ratios for some cases. The method compresses automatically-identified nonrelevant regions at relatively low quality to yield a better compression ratio than standard compression. We show that the method preserves scientific analysis by running statistical tests on it; the resulting probability distribution of analysis results is not statistically distinguishable from the analysis result probability distribution from the original video.

We performed video quality evaluation based on MSD values from tracking diffusing beads. The experiment result suggests that comparing against standard video compression technique H.264, for most compression ratio values, our method gives a better-quality video in terms of the analysis results.

The method extends to other types of microscopy video analysis besides object tracking. The statistical validation method can be modified to apply to other types of analysis. We evaluated quality based on KS test and K-L divergence. If a different metric is desired, the statistical tests can be replaced with the new technique applied to the same population of data.

The correlation-based segmentation method used in our compression technique was verified by analysis of tracking in a fluorescence microscopy videos in our experiment. In Shao et al. (2015), we showed that this segmentation method works for a variety of microscopy video types including fluorescence video, bright field video, fast moving beads video and cell video and associated analysis routines including segmentation.

ORCID

Chong Shao http://orcid.org/0000-0003-4807-4693

REFERENCES

Amer, A., & Dubois, E. (2005). Fast and reliable structure-oriented video noise estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, *15*(1), 113–118.

Bernas, T., Asem, E. K., Robinson, J. P., & Rajwa, B. (2006). Compression of fluorescence microscopy images based on the signal-to-noise estimation. *Microscopy Research and Technique*, *69*(1), 1–9.

Chao, J., & Steinbach, E. (2011, September). Preserving SIFT features in JPEG-encoded images. In *Image Processing (ICIP), 2011 18th IEEE International Conference* (pp. 301–304). IEEE.

Chao, J., & Steinbach, E. (2012, September). SIFT feature-preserving bit allocation for H. 264/AVC video compression. In *Image Processing (ICIP), 2012 19th IEEE International Conference* (pp. 709–712). IEEE.

Chao, J., Chen, H., & Steinbach, E. (2013, September). On the design of a novel JPEG quantization table for improved feature detection performance. In *Image Processing (ICIP), 2013 20th IEEE International Conference* (pp. 1675–1679). IEEE.

CISMM. (2015). Video Spot Tracker. Retrieved from http://cismm.cs.unc.edu/resources/software-manuals/video-spot-tracker-manual/

Cribb, J., Osborne, L. D., Hsiao, J. P. L., Vicci, L., Meshram, A., O'brien, E. T., III, ... Superfine, R. (2015). A high throughput array microscope for the mechanical characterization of biomaterials. *Review of Scientific Instruments*, *86*(2), 023711.

Grois, D., & Hadar, O. (2012). Advances in Region-of-Interest video and image processing. Multimedia Networking and Coding, pp. 76–123.

JVT. (2003, March). Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T rec. H.264-ISO/IEC 14496-10 AVC).

Korshunov, P., & Ooi, W. T. (2011). Video quality for face detection, recognition, and tracking. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *7*(3), 14.

Liu, Y., Li, Z. G., & Soh, Y. C. (2008). Region-of-interest based resource allocation for conversational video.

Menser, B., & Brunig, M. (2000, October). Face detection and tracking for video coding applications. In *Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference* (Vol. *1*, pp. 49–53). IEEE.

Meuel, H., Munderloh, M., & Ostermann, J. (2011, June). Low bit rate ROI based video coding for HDTV aerial surveillance video sequences. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference* (pp. 13–20). IEEE.

Monnier, N., Guo, S. M., Mori, M., He, J., Lénárt, P., & Bathe, M. (2012). Bayesian approach to MSD-based analysis of particle motion in live cells. *Biophysical Journal*, *103*(3), 616–626.

Mukherjee, D., Bankoski, J., Grange, A., Han, J., Koleszar, J., Wilkins, P., ... Bultje, R. (2013). December). The latest open-source video codec VP9-an overview and preliminary results. In *Picture Coding Symposium (PCS), 2013* (pp. 390–393).

Nowak, R. D., & Baraniuk, R. G. (1999). Wavelet-domain filtering for photon imaging systems. *IEEE Transactions on Image Processing*, *8*(5), 666–678.

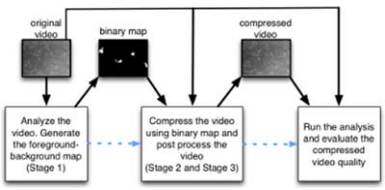
Shao, C., Zhong, A., Cribb, J., Osborne, L. D., O'Brien, E. T., Superfine, R. ... & Taylor, R. M. (2015). Analysis-preserving video microscopy compression via correlation and mathematical morphology. *Microscopy Research and Technique*, *78*(12), 1055–1061.

Sullivan, G. J., Ohm, J. R., Han, W. J., & Wiegand, T. (2012). Overview of the high efficiency video coding (HEVC) standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, *22*(12), 1649–1668.

Van Leuven, S., Van Schevensteen, K., Dams, T., & Schelkens, P. (2008). An implementation of multiple region-of-interest models in H. 264/AVC. In *Signal Processing for Image Enhancement and Multimedia Processing* (pp. 215–225). New York: Springer.

How to cite this article: Shao C, Cribb J, Osborne LD, et al. Analysis-aware microscopy video compression. *Microsc Res Tech*. 2018;00:1–11. https://doi.org/10.1002/jemt.23025

SGML and CITI Use Only
DO NOT PRINT



This article introduces an analysis-aware microscopy video compression method designed for microscopy videos that are consumed by analysis algorithms. Our method can achieve 1,000× compression on certain test microscopy videos.