Analysis-Aware Microscopy Video Compression

**Abstract**

This paper introduces an approach to analysis-aware microscopy video compression method. This method is designed for microscopy videos that are consumed by scientific 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 compress a microscopy video into a much smaller size (TODO: be specific) than modern well-known techniques such as H.264 method does. Comparing against a similar microscopy video compression technique that is designed for yielding the exact same result by analysis algorithms, our method gives more flexibility for a user to control the quality.

**1. Introduction**

The emerge of high-resolution, high-throughput microscopy system such as one described in [2] is driving the growth of video data size produced by scientific experiments. However, the growing of the capacity of storing such data does not follow with the growing of the data size. As a consequence, these video data needs to be compressed before storage.

Various video compression techniques have been invented and standardized in the past decades. Currently the most widely used technique is specified by H.264 standard. As the successor, H.265 has been proposed, standardized and implemented, waiting to replace H.264 in the future [13][14].

H.264 and other popular compression methods are optimized for common video quality metrics such as Peak Signal to Noise Ratio (PSNR). PSNR is one most common measurement on the quality of digital videos. But it is not the best quality metric for all videos including microscopy videos that are feed into analysis algorithms [12]. Based on the goal of an analysis, an algorithm may require extremely high-level preservation of details in part of the video, which exceeds the quality requirement for human visual system (HVS). On the other hand, certain part of the video not relevant to the analysis may be completely ignored by the algorithm. As a consequence, a different measurement of quality should be used to evaluate the quality of a compressed microscopy video. This then leads to the design of a new compression method on microscopy videos.

A novel microscopy video compression method that is designed with respect to a video analysis algorithm should have the following essential features: in compression, it should preserve the information in a microscopy video that is critical to the analysis. It should also do its best to discover the information in the video that is not relevant to the analysis, and greatly compress it. One previous work on exploring single confocal fluorescence microscopy images compression methods is presented in [15]. In their work, they estimated signal to noise ratio (SNR) in microscopy images with the techniques described in [16][17]. The compression was achieved by spatial downsampling, intensity downsampling and wavelet compression.

A correlation-based microscopy compression method has been proposed in an earlier work [1]. It can be considered as an example of the compression technique designed with respect to the analysis. In these experiments, we used a program called Video Spot Tracker [18] that tracks the moving beads in a given video. A correlation-based threshold is used to detect the critical information (foreground) in a video. The foreground is then refined and losslessly stored in the compressed video. The remaining part is compressed by storing the temporal mean of that pixel location. They evaluated their results and show that their method can get at most 100x compression without any change to analysis results. As a comparison, they showed that H.264 compression either yields a smaller compression ratio (lossless) or changes the analysis result (lossy).

The idea of evaluating the quality of a video based on analysis algorithms can be found in [12]. In their paper, 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 compressed video. They discovered that the face recognition and face detection result does not change much until they reach a particular low quality setting. Before that, a compressed video can generate low quality compressed videos that keep the similar face recognition and detection result 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 then proposed that mutual information and blackness are the two computed values that better correlate to the qualities of these analysis results that they can be considered as metrics.

The video compression method described in this paper can be characterized as a Region-of-Interest (ROI) based video compression methods. Previously, ROI video encoding methods has been explored in [4][5][6]. One application of ROI video coding on face detection and tracking is discussed in [7]. Another application on aerial videos is introduced in [8]. Chao et al. discussed the ROI video coding for preserving computer vision visual features in [9][10][11]. For our best knowledge, there is no work done in exploring the ROI video coding in application of microscopy video analysis.

In this paper, we present a microscopy video compression method targeting the preserving of automated analysis results. Differ from the method described in [1] our method does not force the compressed video to have the exact same analysis result as the original video. Instead, to show that the error introduced by compression is no worse than the error introduced by existing noise, we define the video quality as the similarity between the analysis result on the compressed video and the analysis result on the original video under statistical analysis. We run multiple tests followed by statistical analysis. In the case of tracking, we discovered that this quality is highly correlated to the degree of preservation of the pixels surrounding the moving beads.

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 the future work.

**2. Methods**

In designing our compression method, the goal is to have the compressed video contain all the useful information for analysis. In order to do this, the pixels that contain the useful information need to be detected in every video frame.

Due to the fact that the analysis-critical pixels are grouped in regions, the process of determining the analysis-critical region of pixels in every frame can be considered as an image segmentation task. The actual compression process should make use of the segmentation result. It would use different coding scheme or different quality settings for the analysis-critical regions and the remaining regions. So it does its best job to allocate the bandwidth resource for the analysis-critical regions.

We designed our method based on this idea. Our method can be understood as a three-step approach. As in the previous approach, in the first segmentation stage, analysis-critical region in every frame in the video is detected. The method runs a correlation-based approach to decide the important part of the video. As a result, every pixel in every frame is labeled as either foreground or background. This result is stored in a binary map. Following the segmentation stage, the binary map is sent into a compression routine. The compression integrates the segmentation result in its encoding process, so that in any setting for the encoding, the given fixed resource is allocated in a way to ensure that information in the analysis-critical region is well preserved. Inside this stage, we proposed two different variations. They are detailed in section 2.2. After the compression is completed, the resulting compressed video generally has a much smaller size and it is still useful for analysis. However the compression may still introduce error into the analysis result. To address this problem, we designed a post-processing stage to refine the compressed video. The post-processing stage makes use of the noise statistics in the video and refine the video by reproducing the noise that matches the video system characteristics. The post-processing stage is explained in section 2.3.

* 1. **Segmentation Stage**

The goal of the segmentation stage is to accurately detect the regions of pixels in a microscopy video frame that is critical to the analysis. One robust method for this task is described in [1]. In the paper they showed a correlation-based approach. They also designed a refinement stage followed by correlation computation. They applied mathematical morphology followed by erosion to clean up small false positives and dilation to expand foreground region to adjust the correlation-based segmentation result. The reason of performing the dilation in their method is to have a conservative estimation of foreground regions. However in our method only a foreground / background indicator that can be used in later stages is required. Therefore in the segmentation stage of our method, we choose to apply the correlation-based method with the erosion refinement but without the dilation refinement. The next paragraph details the method.



The correlation-based method is for detecting moving objects in a microscopy video. It makes use of noise in a video and the effect Point Spread Function (PSF) TODO: explain in the video acquisition phase. Because of the PSF, every bright pixel is correlated with its neighboring pixels. This does not hold for background noise pixels. To get a measure of this property for every pixel, we compute the Pearson’s correlation score between every pixel and its neighboring pixels. The formula is:

| [Eq. 1]

where xj is the pixel intensity value for the center pixel at jth frame, is the mean pixel intensity value for the center pixel. yij is the pixel intensity value for the neighbor pixel at jth frame; is the mean pixel intensity value for the neighbor pixel.

We compute this value for all eight neighboring pixels for every pixel. We then compute the maximum of all neighboring pixels and use a threshold on this score to determine which pixels are in the foreground. After every pixel has a score assigned to it, we select a threshold such that all pixels whose score are above the threshold are considered foreground pixels. The result is a binary map. The map in general will contain many small groups of false-positive pixels. They are false-positive pixels since their size is smaller than the size of PSF. We remove them with a mathematical erosion operation. Fig. 1 shows one example of the test video frame image and the result binary map cleaned up by erosion. In the next stage the result binary map will be used in compression.

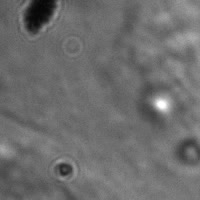
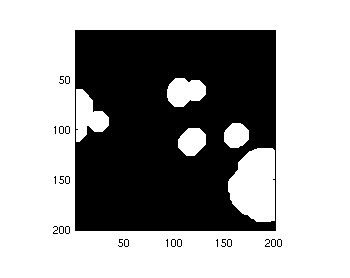
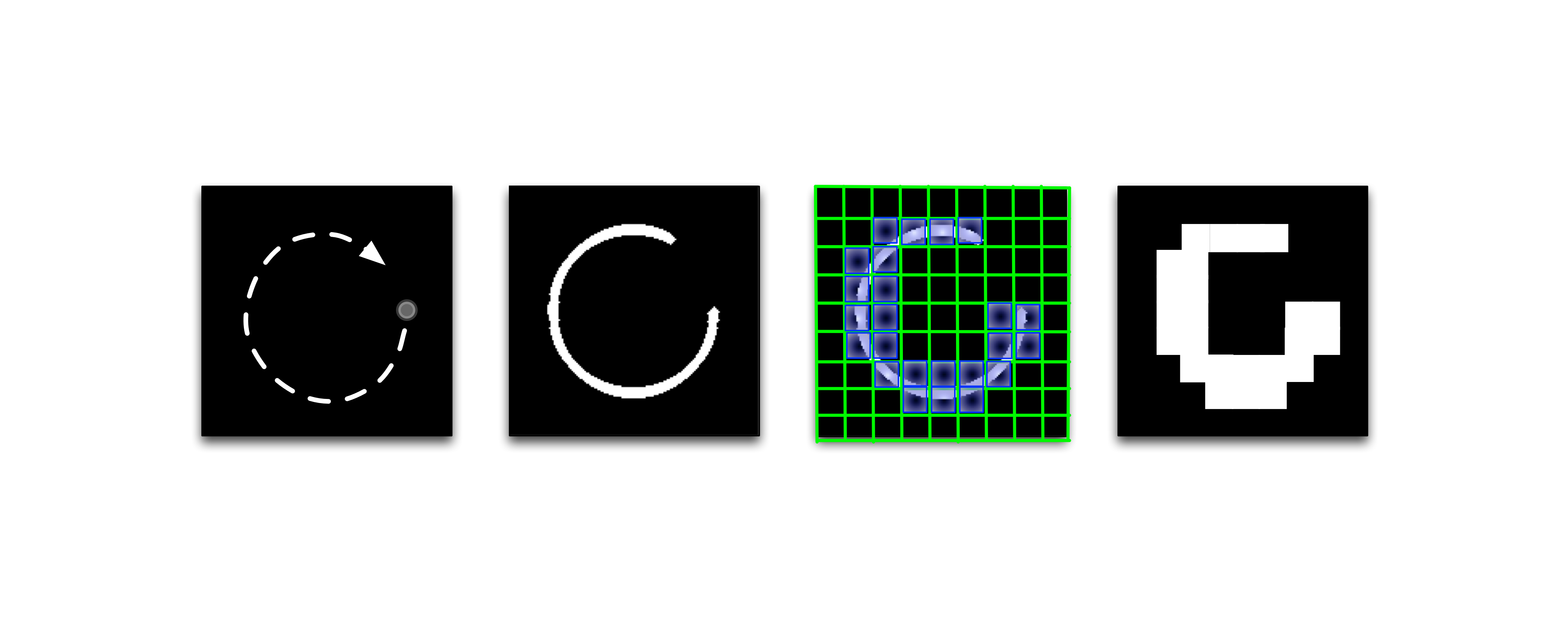
 

Fig. 1: left: sample compressed video frame, right: the segmentation result after erosion refinement for that frame. (TODO: insert the original binary image)

**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 the analysis-critical region information is preserved as good as possible given the constraints. There exist multiple choices in applying existing well-developed video codecs and integrating the analysis-critical map signals in compressing the video data. In developing our system, we explored two paths. The first one is to process the video frames by averaging the “background” pixel values over time based on the segmentation result and then compressed the processed video frames. The result video has many pixel locations that have constant value over time. This type of video data will give good [xxx how much] compress ratio under any common data / video compression techniques. In the experiment we chose to apply H.264 encoding using x264[ xxx version] on the processed video data. But this approach made no strong assumption on the next stage and therefore it provides much flexibility. Tests have been done with three compression software and library: bzip2, jpeg2000 and H.264. The result showed that the three popular compression routines all give a better [xxx how much] compression with variation one comparing against running the compression individually without our approach.

Another approach we designed makes one assumption on the actual compression technique being used. It assumes that the compression is using a block-based prediction-residual compression approach. In particular, our implementation assumed that the next compression technique is H.264. In H.264, the motion estimation unit is based on 16x16 pixel patch (macroblock), and the data reduction is achieved by eliminating the high-frequency terms in each macroblock (quantization). 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 don’t need high quality for blocks that represents background pixels. Therefore our approach assigns different quantization levels to each block based on the segmentation result. As shown in Fig. 2, if in one block there is one or more pixels that is classified as foreground in the binary map, we use a high quality setting (qp 0 or 1), otherwise we assign a higher qp to the block.



a) b) c) d)

Fig. 2. Left to right: a) the starting position of the bead in the video and its moving trajectory; b) the resulting binary map; c) illustration of the macroblocks that covers the frame, the shadowed blocks are marked as foreground; d) the resulting binary macroblock foreground/background labeling map

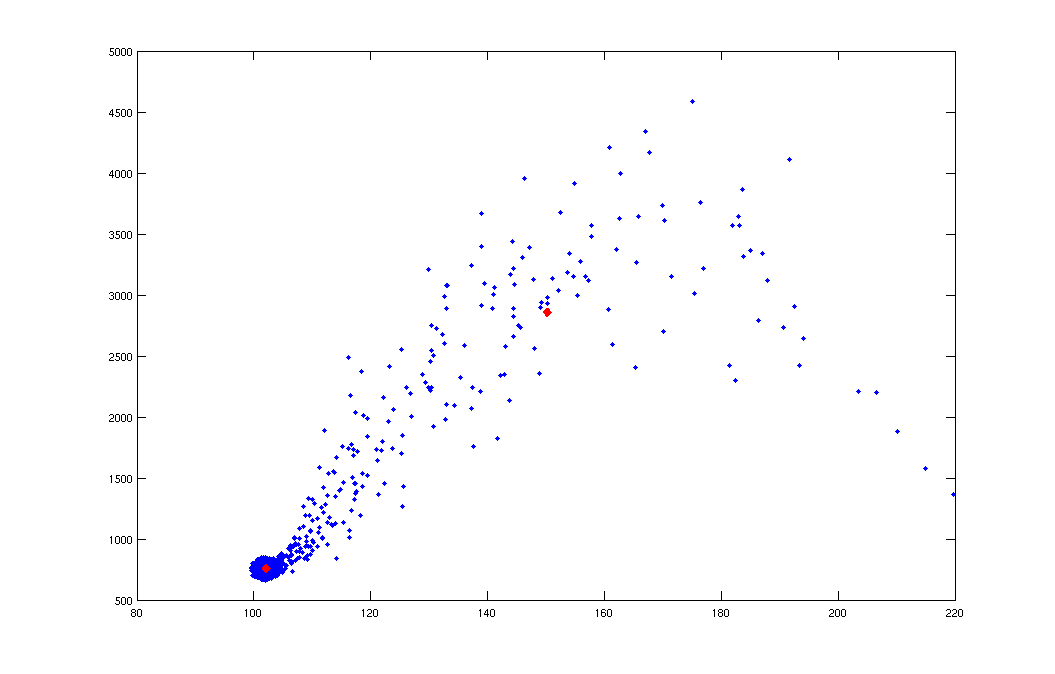
One may also consider combining the two variations: averaging the background and do a customized qp assignment in compression. However, experiments showed that this combination does not yield a smaller compression size. What’s more it even makes the compressed video larger. The reason may be that the 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. This makes the video harder to compress.

**2.3 Post Processing Stage**

Ideally, in the segmentation stage, all pixels that are used for analysis should be classified as foreground. However this is not always the case due to the limitation of the correlation-based segmentation method. There are two possible ways to address this problem. First, one can make a conservative estimation on the segmentation result by either lowering the threshold on correlation-based scores or expanding the foreground regions [1]. The new approach is to estimate the distribution of background noise in the original video, and to add the synthesized noise into the compressed video. This process is the post-processing stage of our method. One can estimate the noise statistics and store the noise parameters. In consuming the video, the noise can be generated and added back into the video in an on-line fashion. Therefore it does not require much additional storage space.

[Eq. 2]

In estimating the noise parameters, we made the assumption that the noise follows the Poisson + Gaussian distribution described in Eq. 2. In assuming a large sample size and simplifying the distribution into another Gaussian distribution with nonzero mean, the only parameter in question is the mean and variance of the distribution. In obtaining the parameters, we used k-means clustering method. By finding the two clusters of the pixel intensity-over time points in the mean-variances space, we use the cluster with lower mean and variance and use its center as the mean-variance of the noise distribution. One sample plot of the pixel intensity over time’s mean vs. variance plot is shown in Fig. 3 where point A is the cluster center being chosen.

****

Variation

Mean

B

A

Fig. 3 mean vs. variation intensity plot with centers of the two-means cluster

**3. Result Evaluation**

Since standard video quality metrics such as PSNR or SSIM does not correlate well with analysis such as object tracking [12]. We seek a better metric for evaluation. Such a quality metric should correlate well with the “true” quality of the video that the scientists really care about. And such a metric is likely to be analysis dependent. For microscopy videos, various types of analysis exist. In this paper, we focus our discussion on the analysis of the tracking on beads attached on cells. In [1], the quality of the video is determined by running the same tracking analysis on the video and only the video with output exactly matches the original video’s analysis result passes the validation.

In our work, we consider the fact that the analysis result on the original video is also affected by the noise existing the previous stages of the experiment pipeline. Therefore the compressed video’s analysis does not have out exactly match it. In stead, we propose a robust statistical-based video quality measurement. It is based on the values derived from the tracking result.

One useful type of data that can be derived from bead tracking results is a curve of mean square displacement (MSD). MSD value is calculated by averaging the squared displacement over the mean position of the movement inside a fixed time window (τ). A sequence of MSD values with increasing time window values contains information about the type of cell motion. By characterizing at the shape of the MSD vs. τ curve, a type of movement can be classified [3].

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 this MSD values together with the MSD values from the tracking on the original uncompressed video. We would like to verify that the error introduced by the compression is no worse than the error introduced by existing noise.

**3.1 Quantitative Tests**

Since we judge the quality of a compressed video by comparing its error with the error introduced by noise in the video, the randomness of noise should be taken into account. To correctly model the behavior of noise and make a comparison, a quantitative approach in preferred. In testing the performance of our methods, we applied two experiments: Two-sample Kolmogorov-Smirnov (KS) test and Kullback-Leibler (K-L) divergence computation. We also plot the MSD mean values across a set of similar videos scaled as multiple of the MSD value for the original video for different compression methods.

1. KS test

In our experiment, the goal is to tell if 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 KS test.

KS test is a well-known technique for testing and giving a confidence level of having two groups of values drawn from two continuous random distributions are actually from the same distribution. Unlike t-test, which mainly tests the difference on two populations’ means, KS test take the shape of the distribution in to account and finds the largest vertical distance between two kernel density plots.

B) 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 non-symmetric: KL-div(P, Q) is generally different from KL-div(Q,P).



Fig.4 flow chart of the experiment steps with synthetic data

**4. Experiments**

We performed two types of experiments to evaluate our methods. The goal is to compare the analysis-aware video compression method against the standard video compression technique (H.264). We included the two variations of the analysis-aware video compression method in comparison. Both synthetic microscopy video data and real microscopy video data are used in the experiments.

For each compression technique, we would like to compare the performance of different compression methods under different bandwidth settings. 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.

**4.1 Experiment on Synthetic Data**

The overall experiment flow on synthetic data is illustrated in Fig. 4. We wrote a program to generate synthetic microscopy video frames. This data generating process is composed by several stages. Firstly, we use a program to simulate bead trajectories. In this experiment we generated 10 bead trajectories. The data was stored as list of x-y pairs, describing the bead positions on every frame in the video. For an 1800-frame video with 10 beads, we had 10 lists with length 1800. With the bead trajectories data, we generated 10 videos that contain beads. All ten videos share the same bead trajectories.

In the second stage, for one beads position on every frame, we used a script to generate a bright 2D Gaussian blob with pre-determined mean intensity and standard deviation values. We place it so it is centered at the given x-y location according to the trajectory data list. The result is a “clean” video without background noise.

The next step is to add noise into the video. In generating a noisy video, we wrote a program to sample values from a pre-determined distribution that models the background noise. Therefore in every video, the background pixels values differ, but they are samples from the same distribution.

The noise distribution is described by Eq. 2. We generated the noise intensity values with one Gaussian plus Poisson distribution with pre-determined λ and σ values [xxx what values?].

The final result is a set of 10 noisy videos. They share the same bead trajectory and they have different background noise. Every video contains 10 beads. Every video has 1800 frames. Fig. 5 (a) shows 1 frame in one of the 10 videos.

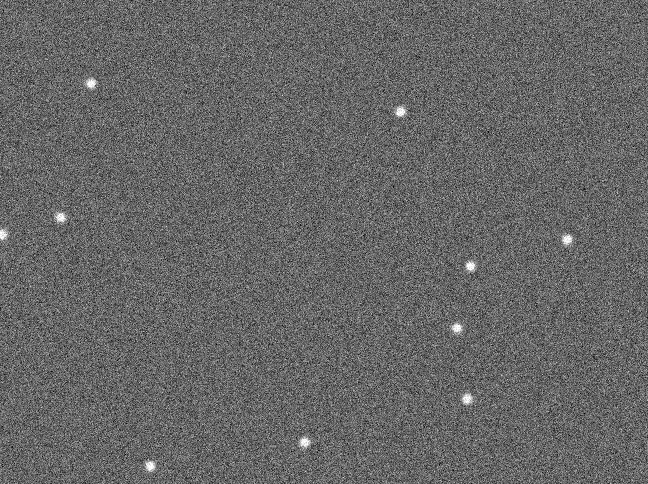
 

Fig. 5 sample video frames, left: synthetic video, right: real video

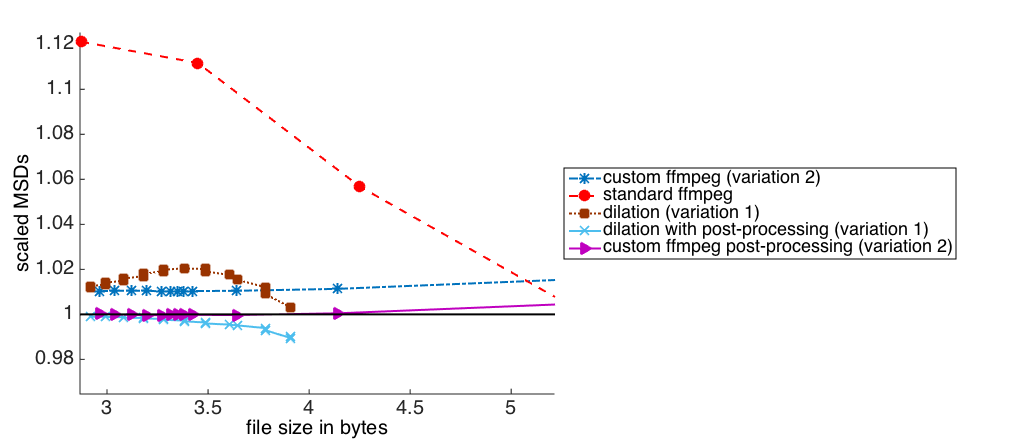
We tested the compression methods with the noisy videos from the data generation process. We compressed every video with 3 methods: our compression approach variation 1 (a), our compression approach variation 2 (b) and the standard H.264 compression (c), each under different settings. In every setting, the parameter we tune are the foreground qp for H.264 compression for (b) and the overall qp for (a) and (c). For (b), the background qp is always set to 51 to achieve the best bitlength saving. [xxx make sure] For every method, we compressed the videos with different parameters to generate a set of compressed videos with different video sizes. We explored the uses of parameters so that the sizes of compressed videos from different compression methods are matched. For each method, we ran the compression with the set of 10 videos to generate a population of compressed videos. Finally for compressed videos using our methods, we ran the post-processing stage on compressed videos with our approach. The compressed video size for the videos after post-processing is considered the same as the compressed ones before post-processing, since the post-processing can be performed when needed.

Finally, we obtained a set of videos we can run analysis with. The analysis we applied is the tracking of beads using video spot tracker [18]. We applied the tracking on the original uncompressed video and all the compressed videos (before post-processing and after post-processing).

Note that by including analysis-aware compression variation 1 in the experiment, we actually also include the method described in [1]. Because in increasing morphology dilation size in the refinement stage described in [1], we eventually reach a large enough dilation size that makes the analysis result exactly the same as the one for original video.

After that we computed the MSD for the tracking results on videos in various sizes. Fig. 6 shows the relationship between MSD values and video compression ratios. The MSD values were scaled as multiple of the MSD values for original video therefore 1.0 means the MSD values are identical to the ones from original videos. Every curve represents the mean MSD values among 10 copies of videos with the same foreground and different noises (noises are on the same level).

The result suggests that two variations of our compression method can both yield a better quality compressed video since the points on the curves are all close to one (in the range between 0.98 and 1.02) While standard H.264 yields a curve far away from 1.0 given the same compression ratio.

One can also notice that the curves for the videos after post-processing are closer to the 1.0 horizontal curve. This shows that post-processing of adding back-noise improves the video quality in terms of analysis.

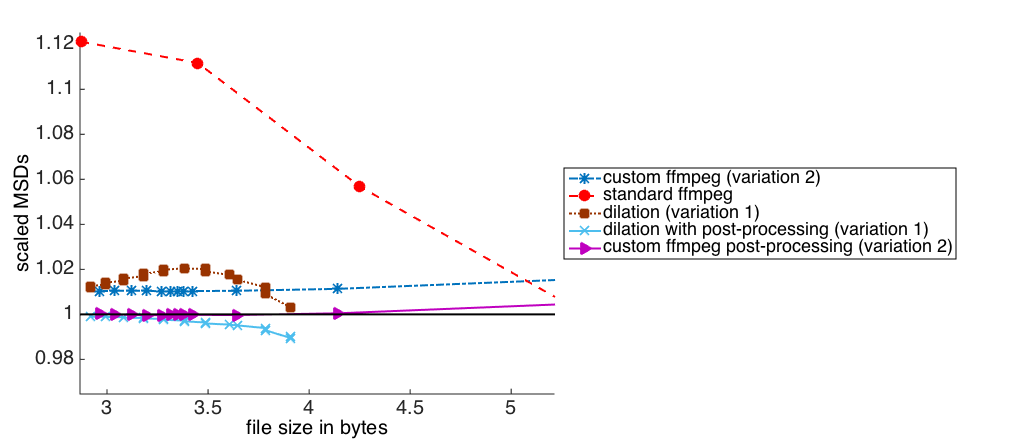
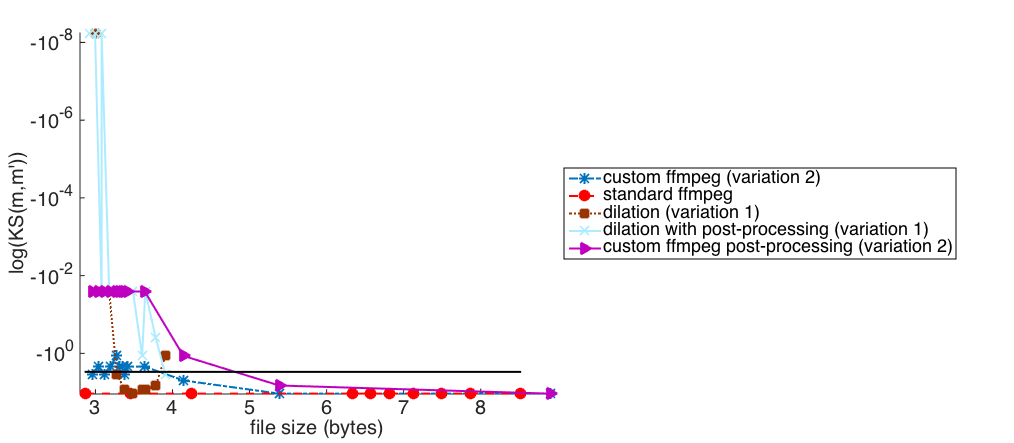
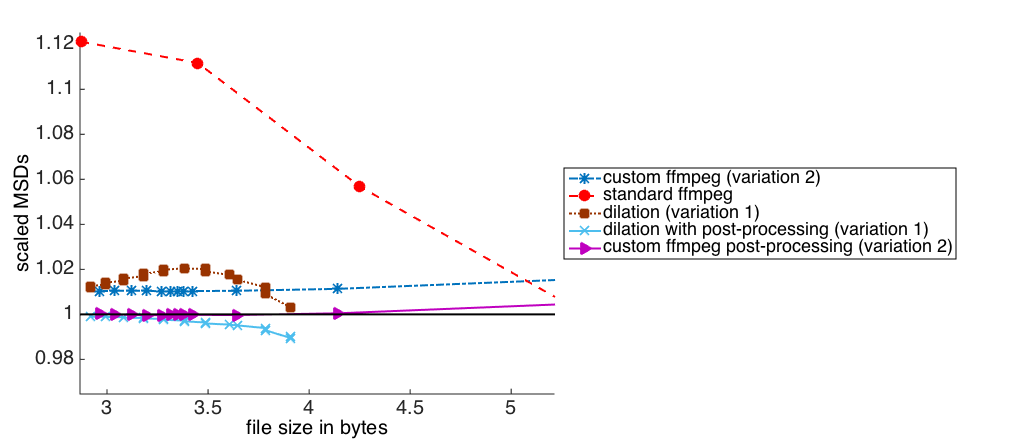


Fig. 6 scaled MSD values vs. compression ratio, for five groups of synthetic videos, note that the horizontal axis is in log scale for display purpose

**4.1.1 KS test**

We performed the KS test on the MSD values. In this experiment the MSD values were not normalized by the MSD values for the original videos. We used one bead’s MSD values across 10 versions of the videos that share the same foreground content. After that, we selected a fixed window size. The left figure in Fig. 7 shows the p values output from KS test for MSD values on videos compressed using our compression approach variation 1 (averaging background), our compression approach variation 2 (custom qp assignment in H.264) and the standard H.264 compression. The curves are p values vs. different compression ratios in log scale. The ks p values are also in log scale and 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 to say that the MSD values obtained from compressed videos are in the different population from the MSD values obtained from the original video.

From the plot, one can see that for the standard H.264 compressed video, the KS test always give the decision to reject the null hypothesis. For our approach, before the post-processing, the curve sometimes goes above the threshold but it also falls below the threshold as the compression ratio increases. For the video compressed with our approach after post-processing, the curves are always above the threshold until it reaches a very high compression ratio.



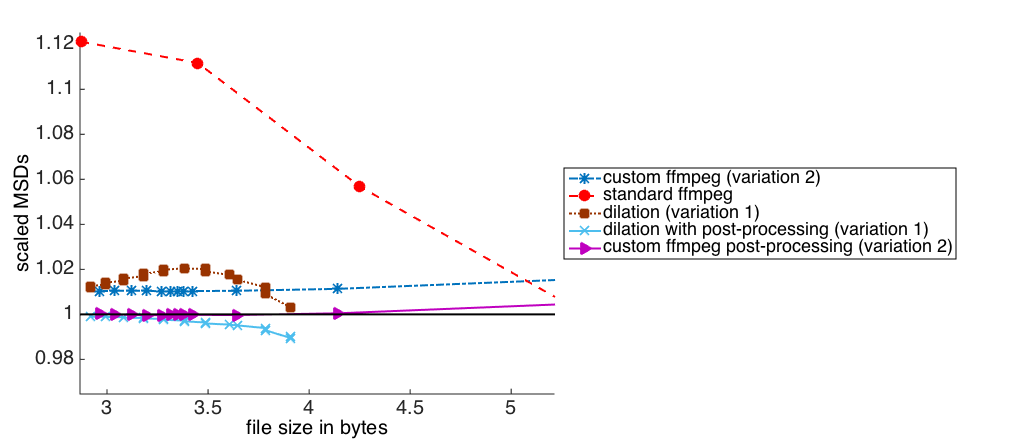
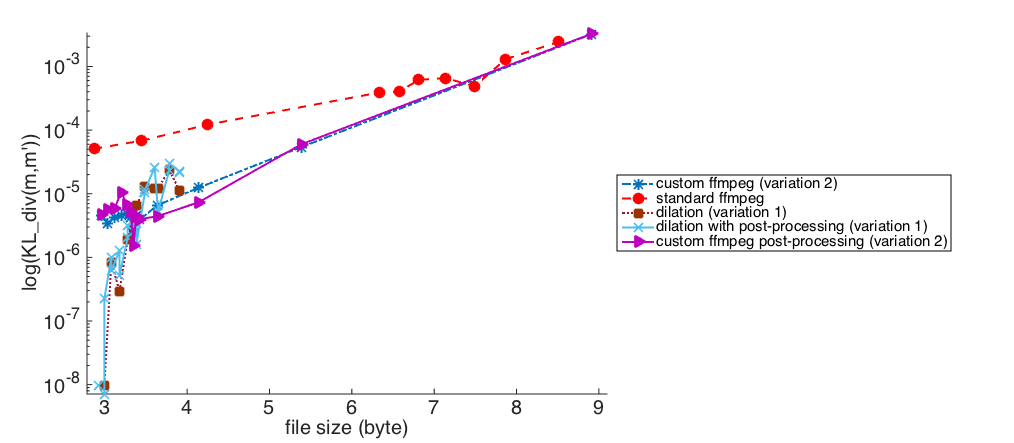
****

Fig. 7 log KS test p values and K-L divergence values vs. compression ratio, for five groups of synthetic videos, note that the horizontal axis is in log scale for display purpose

**4.1.2 K-L divergence**

We performed the computation of K-L divergence values on the same data set as the one described on 4.1.1. The result is shown on the right figure in Fig. 7. Note that in this case lower K-L divergence suggests a smaller distance between the compressed video MSD value population and the original video MSD value compression.

In this experiment, all compression videos with our method (before and after post-processing) gave similar good result for compression ratio smaller than 148 (5 on the horizontal axis), better than standard H.264. For our compression approach variation 2, we were able to obtain a larger compression ratio, but for a very large compression ratio, our compression approach variation 2 does not give a better performance compare to standard H.264.

**4.2 Experiment on Real Data**



Fig. 8 flow chart of the experiment steps with synthetic data

To prove that our method is practical, we also did experiments on real data. In real data, it is impossible to get the true bead trajectory and generate multiple copies of the same bead trajectory and difference background noise. We got around this by dividing 1 video into 10 parts and perform tracking on it. By doing this, there is no significant background noise property change across the videos in the test set. The experiment process is illustrated in Fig. 8.

The scaled MSD values for various compression methods vs. video compression ratio plot for real video data is given in Fig. 9. For real data, our compression approach variation 1 does not perform better than the standard technique (H.264) for compression ratio greater than about 200 (5.3 on the horizontal axis), even after post-processing. But our compression approach variation 2 always gives a better result than standard technique (H.264).

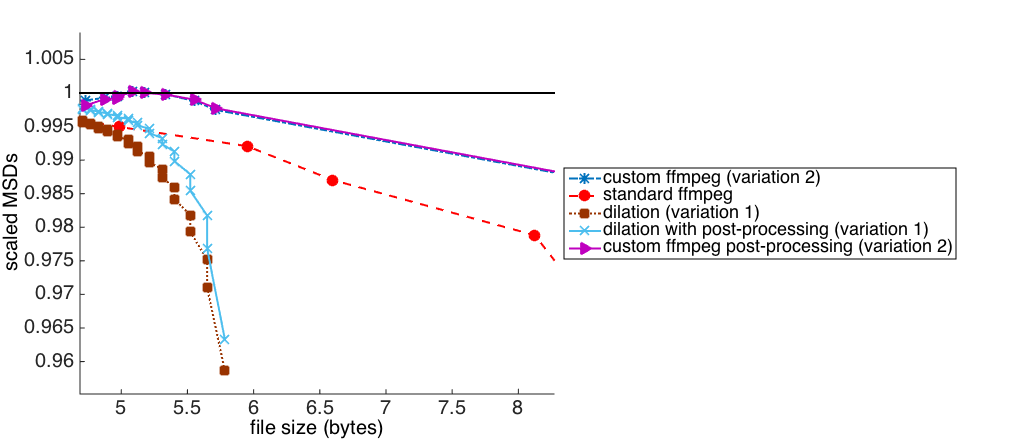
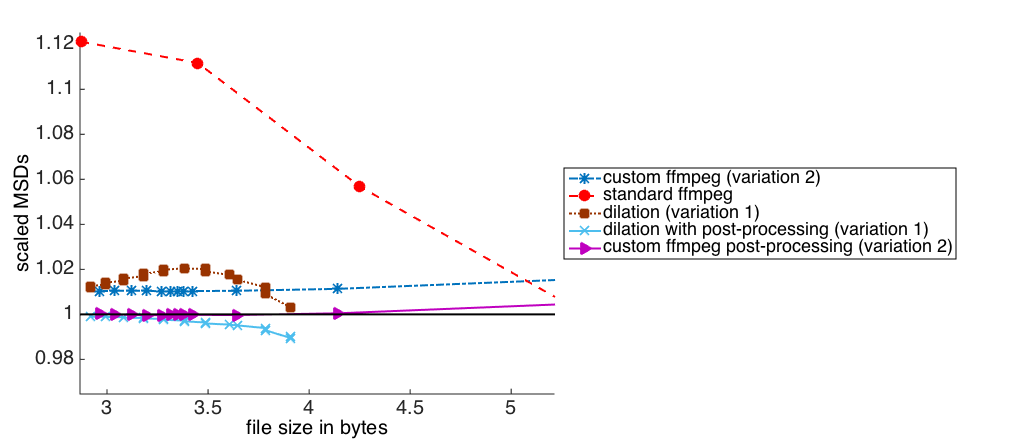
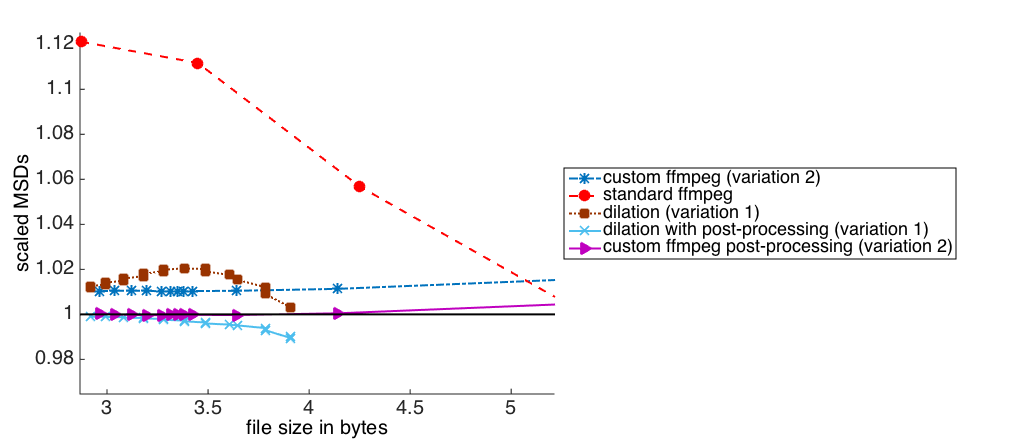
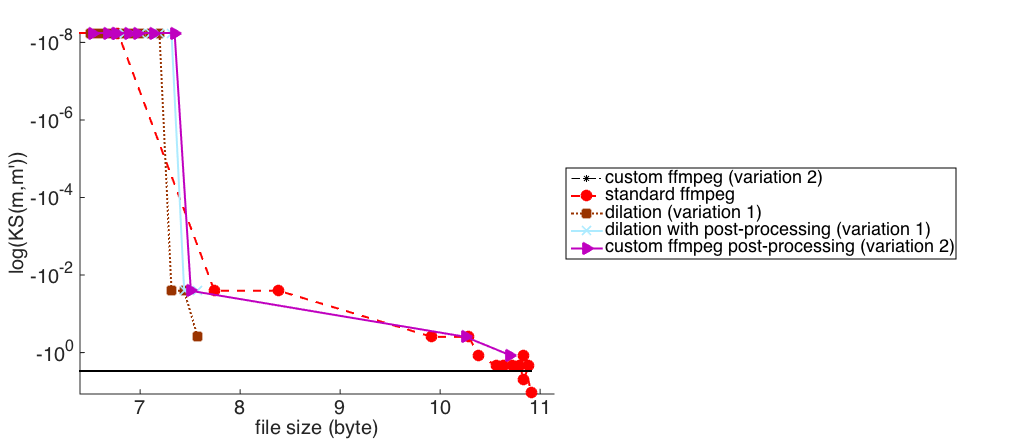


Fig. 9 scaled MSD values vs. compression ratio, for five groups of real videos, note that the horizontal axis is in log scale for display purpose

****

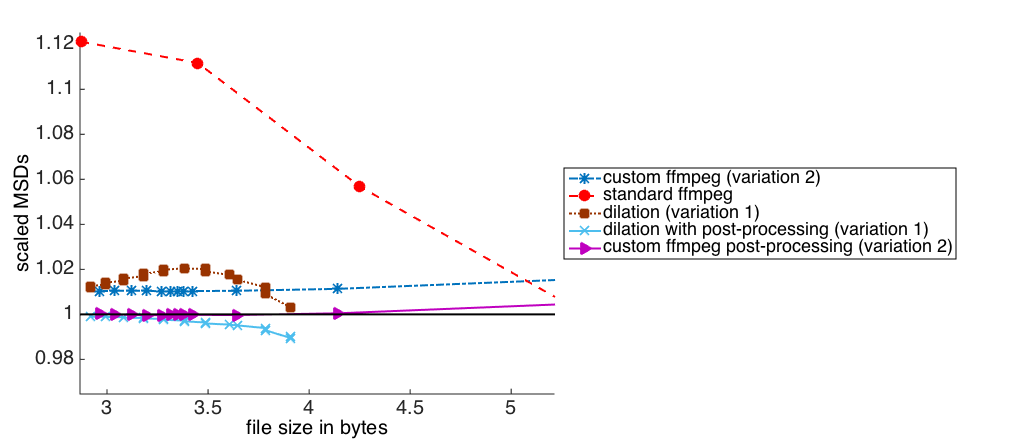
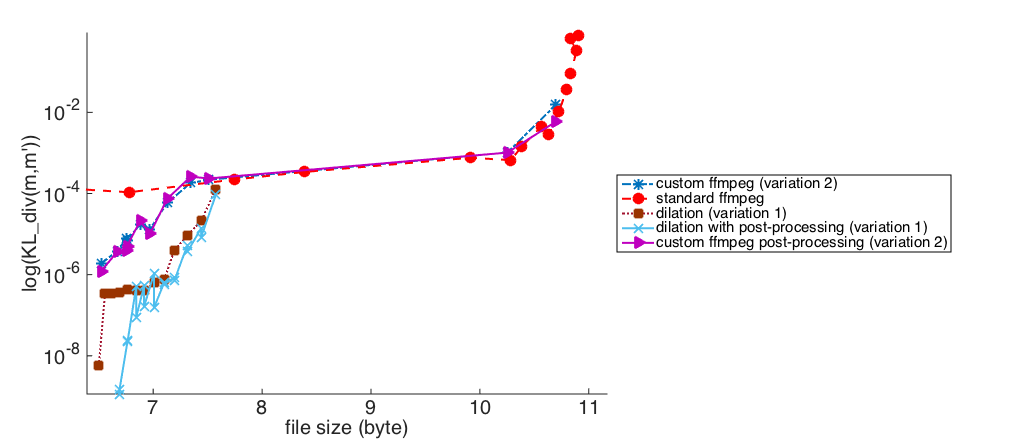
****

Fig. 10 log KS test p values and K-L divergence values vs. compression ratio, for five groups of real videos, note that the horizontal axis is in log scale for display purpose

Fig. 10 shows multiple compression methods’ result MSD values’ KS test p value (in log scale) curves and K-L divergence value curves for various video compression ratios (in log scale). For real data, K-L divergence does not show a great difference among different compression methods. In KS test, the p values for the videos compressed by standard H.264 starts to drop below the threshold after it reaches a certain compression ratio, while our compression method variation 2 goes above the threshold under the same compression ratio (our compression method variation 1 cannot reach that large compression ratio).

**4.3. Discussion**

The correlation-based segmentation method used in our compression technique was only verified with the analysis of tracking in our experiment. But in [1], we showed that this segmentation method works for many microscopy video types including fluorescence video, bright field video, fast moving beads video and cell video and associated analysis routines including segmentation.

In the experiment on synthetic data, since we know the true position of the beads, we could compare it with the tracked bead locations from different videos. By doing this we realize that adding noise will make the MSD higher. Fig. 11 shows an example: in this plot of MSD value curves vs. different τ values, the compressed video MSD curves (yellow) is closer to the MSD curve for the true bead location (black). By considering this, the compression is actually given the denoising on the original data, to make it more close to the underlying truth. In certain cases, it may be preferred.

**5. Conclusions and Future Work**

In this paper, we introduced a microscopy video compression method. Our method is based on finding the analysis-critical information in a microscopy video and compressing the analysis-critical information with high quality. At the same time our method compress the remaining part of the video in relative low quality to yield a better compression ratio. [xxx rewrite]

We tested our compression framework with a specific setting: a correlation-based segmentation method borrowed from [1]. We proposed a video quality evaluation based on MSD values on the tracking results. Experiment result suggested that, comparing against standard video compression technique H.264, for most compression ratio values, our method gives a better quality video in terms of analysis result.

The method should be able to be extended to other type of microscopy video analysis rather than object tracking. However the statistical validation method may be different for each type of analysis. Further study should be done to easier determine if a statistical validation method is suitable for a given type of analysis.

We proposed the quality evaluation based on statistical tests. If a novel metric can be found and directly applied, then statistical tests can be replaced since they involve a population of data.

**References**

[1] Chong Shao, Alfred Zhong, Lukas D. Osborne, Jeremy Cribb, Tim O’Brien III, Ketan Mayer-Patel, Russell Taylor II, and Richard Superfine. “ Analysis-Preserving Video Microscopy Compression via Correlation and Mathematical Morphology”, MRT

[2] Cribb, Jeremy, Lukas D. Osborne, Joe Ping-Lin Hsiao, Leandra Vicci, Alok Meshram, E. Tim O’Brien III, Richard Chasen Spero, Russell Taylor II, and Richard Superfine. "A high throughput array microscope for the mechanical characterization of biomaterials." Review of Scientific Instruments 86, no. 2 (2015): 023711.

[3] 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.

[4] S.V. Leuven et al. An implementation of multiple region-of-interest models in H.264/AVC

[5] Yang Liu et al., Region-of-interest based resource allocation for conversational video communication of H.264/AVC

[6] Dan Grois and O.Hadar, Recent Advances in Region-of-interest video coding

[7] Bernd Mensor Face ROI paper

[8] Holger Meuel Arial ROI paper

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

[10] 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 on (pp. 709-712). IEEE.

[11] Chao, J., Chen, H., & Steinbach, E. (2013, September). On the design of a novel JPEG quantization table for improved feature detection performance. InImage Processing (ICIP), 2013 20th IEEE International Conference on (pp. 1675-1679). IEEE.

[12] 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.

[13] JVT, “Draft ITU-T recommendation and final draft international

standard of joint video specification (ITU-T rec. H.264-

ISO/IEC 14496-10 AVC),” March 2003.

[14] 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.

[15] Amer A, Dubois E. 2005. Fast and reliable structure-oriented video noise estimation. Circuits and Systems for Video Technology, IEEE Transactions on. vol.15, no.1, pp.113-118.

[16] Bernas T, Asem EK, Robinson JP, Rajwa B. 2006. Compression of fluorescence microscopy images based on the signal-to-noise estimation, Microsc. Res. Tech., 69: 1–9. doi: 10.1002/jemt.20259

[17] Nowak RD, Baraniuk RG. 1999. Wavelet-domain filtering for photon imaging systems. IEEE Transactions on Image Processing. vol.8, no.5, pp.666-678.

[18] Video Spot Tracker: <http://cismm.cs.unc.edu/resources/software-manuals/video-spot-tracker-manual/>

[xxx more references, some references of microscopy video compression and xxx]