Response to editor comments:

I've spend a couple weeks implementing and then troubleshooting some code to test the effects of motion blurring on the particle area measurements using circular particles while taking the individual image processing algorithms of each instrument into account. While I was successful in implementing this I found that applying a 50% threshold (2DVD uses a flat threshold determined via calibration) to the motion-blurred circular particles resulted in the complete removal of the motion blurring effects. At first I found this rather confusing, but, as I thought about it more, I came to realize that this was not a coding issue but a fundamental flaw with this approach to examining the motion blurring. Motion blurring of a step function effectively replaces the step with a linear gradient. The motion blurring algorithm I had implemented (specifically, Korein and Badler 1983) assumed the object being blurred had a brightness of one everywhere and that the background had a brightness of zero (i.e., the edge of the circle was a step function). Taking the 50% threshold of this linear gradient would place the edge of the motion-blurred particle at the same location as the edge would be had the particle been stationary at its position at the midpoint of the camera exposure period.

After some consideration, we decided it best to simply remove the discussion of particle size measurements as we feel a proper test of the effects of motion blurring on particle size measurements will a) likely require considerable additional work beyond what has already been done prior to and during this review process and b) probably strongly benefit from empirical testing using the actual instruments involved. As the bulk of this paper was focused on the performance of the particle shape measurements, we feel that there is still considerable merit to completing this review process after removing the discussion of particle size measurements. Since the portions of the manuscript to which comments 1 and 2 refer have been removed from the manuscript, we will forgo responding to them. Our response to comment 3 is included below, however.

In addition to removing text relating to the size measurements and changes made in response to comment 3, the other main change to the text is in the methods section, where we have updated the text to reflect that PIP image processing is now applied to the raw PIP images prior to applying the shape-fitting algorithms. This change had been made to the code during a previous round of revisions, but, for whatever reason, the text was not updated to reflect the change.

Lines 184 – 202:
"For the emulation of both MASC and 2DVD as well as the implementation of the tensor method, the PIP images taken from the AVI files undergo image processing matching that of the PIP (Newman et al, 2009; L. Bliven, 2022, personal communication). First, the 8-bit image is passed through a Prewitt edge detection filter and the results are threshold such that all pixels with a value of at least 26 are assigned a value of one and all other pixels are assigned a value of zero. This 2-bit image is then dilated twice using a three-pixel by three-pixel kernel where all elements are set to one. At this point all interior holes are filled with values of one and the resulting 2-bit image is eroded three times using the same kernel as was used in the dilation step. The particles in the resulting image are composed of pixels that have a value of one.

After the image processing, particles are located by matching the particle pixels to the particle positions, as identified in the PIP data files, by using a region grow approach, whereby a coherent region is grown from a single point by checking its eight neighboring pixels for inclusion in the region and then repeating the process with each newly included pixel. The region grow approach is initially attempted at the PIP-

determined particle centroid. For some particles with large concave regions, the PIP-measured particle center can correspond to a non-particle pixel. In these instances, an attempt is made to locate a new starting pixel for the region grow technique by searching the following (x,y) coordinate offsets in the given order: (-1,0), (1,0), (0,1), (-1,1), (1,1), (0,-1), (-1,-1), (1,-1). If a particle pixel is not located at any of these center offset positions, the process is repeated after multiplying the offsets by a factor of two; in cases where a particle pixel remains elusive, the process is repeated with multiples of three and then four. If no particle pixels are located after checking these 32 offset positions, the particle is considered invalid and is ignored. Additionally, only particles that appear both in the particle tables (i.e., the PIP files ending in `_a_p.dat') and in one of the two velocity tables (i.e., the PIP files ending in either `_a_v_1.dat' or `_a_v_2.dat') are considered here as the variables contained in these files are necessary when performing the emulations."

Additionally, I've changed the color table for the 2D histogram figures to make them more colorblind friendly (Figs 1, 3, 4, and 8).

3. Artificial cap and discussions around Fig 7 and Fig 8

It is suggested that we perform an additional statistical analysis of the effects of pixilation on the perimeter stretching factor.

As with many aspects of replicating algorithm functionality in this study, this has become far more complicated and uncertain than it seemed at the outset. The crux of the issue is in understanding how the IMAQ software package actually computes perimeter. Again, the frustrations of poor documentation continue. What documentation does exist for the IMAQ perimeter measurement states that IMAQ subsamples the pixels before measuring the perimeter in order to produce a smoother outline. Our interpretation of this was that the pixels were subdivided into subpixels and the perimeter was then drawn using the corners of these subpixels, this is apparently wrong though (in hindsight, our interpretation would be better described as oversampling). I managed to dig up a couple forum posts where users were asking the developer how the algorithm works and the answers, despite still being annoyingly vague, are somewhat illuminating. Our new understanding is that "subsample" means that IMAQ ignores some pixels along the edge of the particle when drawing the perimeter. In the example given in one of the forum posts (screenshot and link below as Figure 1), IMAQ skips every other pixel when computing the perimeter and assigns the location of the edge based on one corner of each pixel (how that corner is selected is anyone's guess; my guess is that it is the pixel corner that is farthest from the particle center). To make things even more troublesome, the number of pixels skipped appears to be a function of the size of the particle; what metric is used for the particle size and how that size relates to the number of pixels skipped is left to our imaginations it would seem.

Regardless, I attempted to test the sensitivity of the perimeter stretching factor to the pixilation of a perfect circle using the path mapped out by IDL's contour procedure as the perimeter. The circles are generated based on the distance of each pixel center from a randomly perturbed center location and include spatial anti-aliasing to more realistically reflect how a circular particle would appear. As I was unsure of whether it would be better to have IDL draw the contour at a value of 1.0 or 0.5, I tried both and was troubled to find that there was a fairly large sensitivity to this selection (see Figure 2). With a value of 1.0, the pixelation effects could be making the perimeter stretching factor smaller or larger, depending on the circle radius. A value of 0.5, however, suggests the pixelation always makes the perimeter stretching factor larger. If the results of both were similar to the results using a contour value

of 0.5 (dashed line), I would be more open to agreeing with you that the pixilation was the dominant factor, but given that the 1.0 contour value results in a strong dependence on radius and the vagueness around the subsampling that IMAQ does when computing perimeter, I'm not comfortable with making that statement. That said, there is a lot more to this than first appeared and I've added some brief text to the manuscript as an explanation. As for the pixilation vs underlying shape, I've left it as "Small increases in perimeter, such as this, can be introduced by a few very small deviations of the particle edge from a perfect circle as well as by the inability to perfectly represent a circle using square pixels (i.e., pixelation effects)."

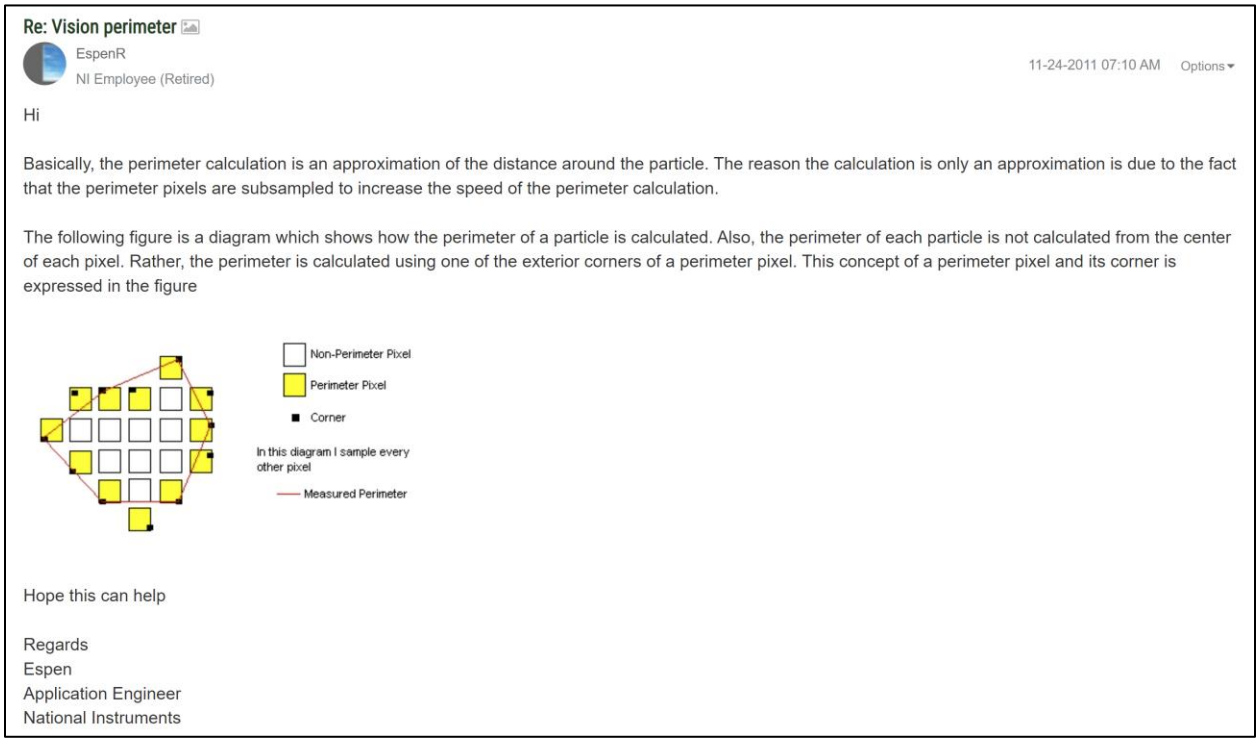

Figure 1. Screenshot of forum reply explaining IMAQ perimeter calculation. https://forums.ni.com/t5/Machine-Vision/Vision-perimeter/m-p/1787588#M33709

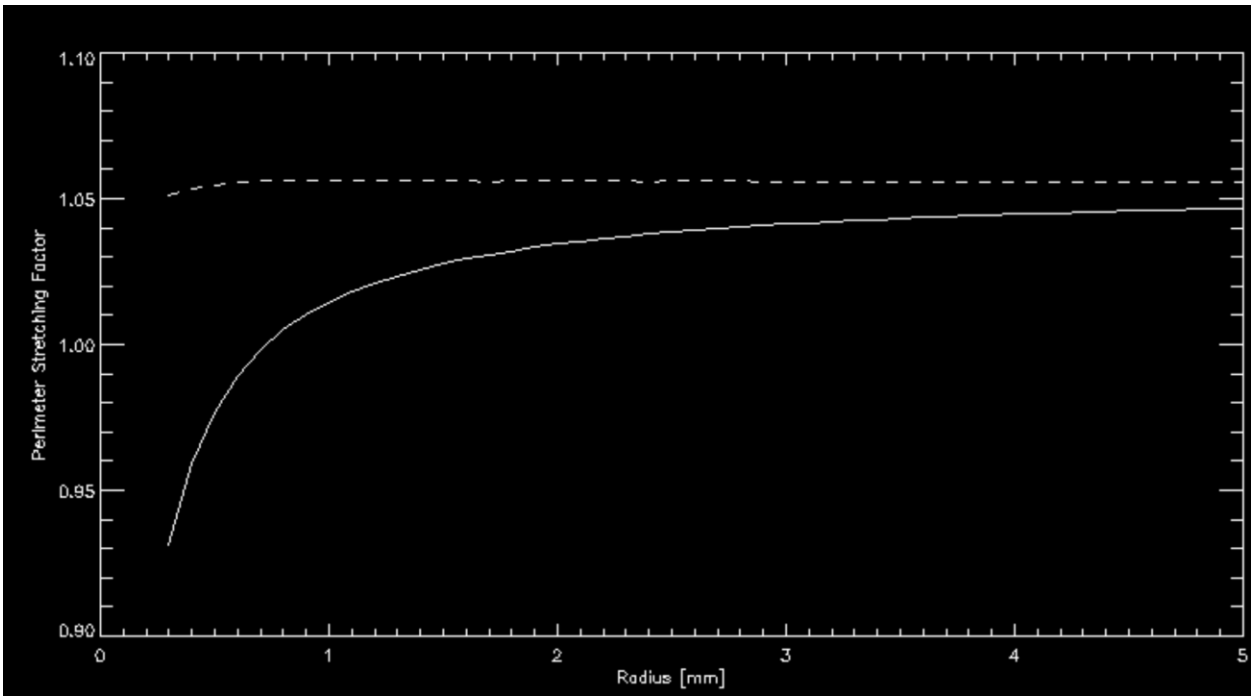

Figure 2. Results of testing pixilation effects on perimeter stretching factor when taking the perimeter to be the IDL contour along a pixel value of 1.0 (solid) and along a pixel value of 0.5 (dashed).

Here are the relevant changes:

Lines 106 – 110:
"Based on online forum discussions[1], it appears that IMAQ computes the particle perimeter from a subset of the edge pixels with the number of edge pixels skipped between each perimeter calculation pixel being a function of particle size. Furthermore, IMAQ uses one corner of each perimeter calculation pixel rather than the pixel center. Unfortunately, we were unable to determine either the function relating particle size to number of skipped pixels or the corner selection method."

The footnote reads: "See the response by user EspenR, an NI application engineer, to the thread "Vision perimeter" on the NI forums: https://forums.ni.com/t5/Machine-Vision/Vision-perimeter/m-p/1787588\#M33709, accessed 29 August 2022."

Lines 350 – 355:
"In theory, it should be possible to determine the contribution of pixelation effects on the PIP shape fitting algorithm; however, as already mentioned, the details of the IMAQ perimeter calculation are not well documented. Additionally, attempting to quantify the contribution from pixelation by measuring the length of a contour around the particle rather than attempting to emulate the IMAQ perimeter calculation resulted in a high sensitivity to the pixel value at which the contour was placed. As such, we have decided to leave the pixelation effects and shape deviations as an unknown in terms of their contributions relative to one another in the PIP shape fitting algorithm."