AN ALTERNATIVE ALGORITHM TO CALCULATE THE BIOSPECKLE SIZE IN COHERENT LIGHT SCATTERING EXPERIMENTS*

D. CHICEA

Physics Department, “Lucian Blaga” University, Dr. I. Ratiu Street, No. 5–7, 550024, Sibiu, Romania,
E-mail: dan.chicea@ulbsibiu.ro

Received September 26, 2007

In a light scattering experiment the far interference field carries information regarding the scattering centers. The biospeckle size is currently calculated as the normalized autocovariance function of the intensity pattern. A different algorithm is proposed to calculate the biospeckle size and examples of using it are presented.

Key words: coherent light scattering, biospeckle.

1. INTRODUCTION

If medium having scattering centers randomly distributed, is the target for a laser beam, an un-uniformly illuminated image is obtained, currently named speckled image, having a statistical distribution of the intensity over the interference field. The speckled image appears as a result of the interference of the wavelets scattered by the scattering centers (SC hereafter), each wavelet having a different phase and amplitude in each location of the interference field. The image changes in time as a consequence of the scattering centers complex movement of sedimentation and Brownian motion giving the aspect of “boiling speckles” [1], [2].

If the speckled image is observed in free space it is named objective speckle while if it is observed on the image plane of a diffuse object illuminated by a coherent source it is named subjective speckle in [1]. The review paper [2] classifies the two types of speckled images as far field speckle and image speckle. In this work the objective speckle, respectively far field speckle is considered.

It was often pointed out in the literature that the speckle parameters like size, contrast, intensity and polarization carry information on the scattering media.


Dynamical speckle analysis has become a current method to characterize the dynamic behavior of scattering medium such as flow, sediment and Brownian motion. The motion of the speckle field was analyzed by correlometric methods [3–5] or by laser speckle contrast analysis [6, 7]. The speckle size can be used to measure the roughness of a surface [8–10] or to assess the thickness of semi-transparent thin slab like in [11]. Most of the above mentioned experiments use the backscattered speckle configuration. In papers like [12] a different optical set-up is used to measure the correlation function in the near field, and show the near-field speckle dependence on the particles size. The work reported in [13, 14] uses a transmission optical set-up to measure the far field parameters like contrast and speckle size. The transmission type of setup was used in the work reported in this paper, as well.

The work presented in this paper was done to develop and test an alternative, faster algorithm to calculate the speckle size. The algorithm and the procedure to test the algorithm are presented further on. Finally an example of using the algorithm to investigate the speckle size variation with the SC concentration is presented.

2. THE SPECKLE SIZE CALCULATION PROCEDURE

We consider a simple experiment consisting of a laser source, a cuvette containing the suspension, located within the coherence length of the laser source and a frame grabber with CMOS or CCD and a PC to record the frames captured by the capturing device.

We consider \( I(i,j)=I(x_i, y_j) \) to be the intensity recorded by the cell \((i,j)\) of the CMOS, hence by the pixel \((i,j)\) of the array of pixels the image consists of.

In [1], [13], [14] and [15] the average speckle size is calculated as the normalized autocovariance function of the intensity speckle pattern got in the observation plane:

\[
c_l(\Delta x, \Delta y) = \frac{\text{FT}^{-1}\left[\text{FT}[I(x,y)]^2\right] - \langle I(x,y) \rangle^2}{\langle I(x,y)^2 \rangle - \langle I(x,y) \rangle^2}
\]  

(1)

where FT is the Fourier transform, \(< >\) is a spatial average, \(c_l(\Delta x,0)\) is a horizontal and \(c_l(0, \Delta y)\) a vertical profile. The \(c_l(\Delta y, \Delta y)\) function defined in (2) has a 0 base and the width of the function provides a reasonable measure of the speckle size, as stated in [15].

In this paper a different approach is proposed to calculate the speckle size. Each vertical intensity profile is extracted as a series. For each vertical profile of a frame the normalized autocorrelation function [16] was calculated as:

\[
A_s(s) = \frac{\langle I(n,j)*I(n,j+s) \rangle}{\langle I(n,j)*I(n,j) \rangle}
\]  

(2)
where the angle brackets denote averages over the coordinate $y$, $n$ represents the number of the vertical profile and $s$ is the autocorrelation distance. The average was subtracted from each series before calculating the normalized autocorrelation function using eq. (2).

The speckle size for $n$-th profile, (the profile speckle size) is defined as the value of the $s$ (pixel number or distance, if multiplied by the pixel size on the CMOS) where the autocorrelation function decreases to $1/e$. An average of the profile speckle sizes is calculated for each image and we define this average as the average speckle size for that particular image.

The algorithm was tested to verify if the calculated speckle size is a reasonable measure of the actual speckle size. For this purpose different types of speckle images were generated, having both a cosine and an exponentially decreasing profile from the center to the margin of each “speckle”, with different calculated speckle sizes, having the speckles placed either in rows and columns or randomly over the calculated image.

![Fig. 1. – A generated speckled image having the speckle size of 35 pixels.](image)

Each generated figure had a resolution of 1000x1000 pixels, 24 bits color depth bitmap. Figure 1 presents a generated speckle image having the “speckles” randomly distributed over the area, each one having a size of 25 pixels and Figure 2 the same kind of image with the speckle size of 70 pixels. Both Figures presented here have an
exponentially decreasing profile from the center to the margin of each “speckle”. Figure 3 presents a vertical profile on Figure 1 and Figure 4 the autocorrelation function for the profile in Figure 3. For this profile the profile speckle size is 13.75 pixels.

Fig. 2. – A generated speckled image having the speckle size of 70 pixels.

Fig. 3. – A vertical profile on the bitmap in Fig. 1.
We notice that the circular “speckles” are partially overlapped, which simulates well the real image, where the maxima are not equidistant, as in an image where the speckles are generated on rows and columns. The speckle size and the error in assessing it were calculated for different radius values used in generating the images. The results are presented in Fig. 5, where the circles are the calculated speckle sizes, with the estimated errors.

The linear regression equation is $y=0.4130x+0.0101$. 

Fig. 5. – The calculated speckle size, with error bars, versus the generated speckle radius.
The line represents the linear regression, described by equation (3):

\[ S_{ps} = 0.4130 \cdot r + 0.0101 \]  \hspace{1cm} (3)

In (3) \( S_{ps} \) stands for the calculated speckle size and \( r \) for the radius of the generated "speckle". As the radius increased the error in calculating the speckle size increased, because the size of the bitmap remained constant, \( 1000 \times 1000 \), but the number of generated speckles was consequently decreased, otherwise the degree of overlapping turned to be too big.

When the algorithm was tested on generated speckles having a cosine decreasing profile from the center to the margin of each "speckle", the linear regression equation is described by (4), which is very close to equation (3):

\[ S_{ps} = 0.4213 \cdot r + 0.0095 \]  \hspace{1cm} (4)

Examining the images presented in Figures 1 through 4, and the images of the other generated speckle images, not presented here, together with the plot in Fig. 5, we can conclude that the procedure presented in this work, although it is less time consuming, provides not only a reasonable measure of the speckle size but the error in measuring it, as well.

The procedure that was used in this work is apart of the procedures described in [13–15] primarily because it is a lower cost procedure. In [13–15] a CCD with an analog to digital converter was used. In this work a simple CMOS camera was used, connected to a PC through an USB port. The acquisition was done on 24 bits rather than on 10 bits. The images were acquired either as bitmaps or as uncompressed AVI. The program that was written to perform the calculations, as described above, first opens the conventional graphic formats and extracts the intensity array \( I(x,y) \). Another difference lays in the fact that consecutive images extracted form a recorded movie are processed and averaged, rather than a single image.

Finally, the algorithm that is proposed in this paper is less computer time consuming as it does not require the calculation of the Fourier transform on the \( 640 \times 480 \) array twice (the elapsed calculation time goes as \( (640 \times 460)^2 \)), but the autocorrelation function for 480 vectors of 640 elements each (the elapsed calculation time goes as \( 480 \times 640^2 \)). In addition the procedure provides, together with the average speckle size, the error in speckle size calculation, as the standard deviation of the set of profile speckle sizes, as defined above.

It should be noted thought that the computation time difference is negligible if one bitmap is processed, but becomes significant when processing big resolution bitmaps (more than \( 10^6 \) pixels), movies, or sets of movies, in batch mode. Moreover, as technology develops, faster and higher resolution CCDs will be commercially available and the gain in computation time might become appealing in developing an online, automatic procedure for monitoring low SC concentration, as suggested at the end of Section 3.
3. RESULTS OF SPECKLE SIZE ANALYSIS

Moving to the real world, we recorded images of a typical far field speckle, using a CMOS frame grabber, during an experiment having diluted milk as target. A frame is presented in Fig. 6 and a profile in Fig. 7. It should be noted that the resolution is 640×480 pixels, smaller than 1000×1000 used for testing the algorithm, which explains the apparent bigger oscillation period in Fig. 7 than in Fig. 3.

Using the algorithm previously described, the speckle size variation with the scattering centers concentration (milk fat particle for this example) was analyzed and Fig. 8 presents the results. Details on the experimental setup are presented in [17]. Other results, on work done on diluted magnetic fluid, are presented in [18].
4. CONCLUSIONS

A novel algorithm to calculate the biospeckle size is proposed and described in detail. The novel algorithm is essentially different from the conventional algorithm as it does not require the calculation of the Fourier transform on the \( mxn \) array twice, where the elapsed calculation time goes as \( (mxn)^2 \), but the autocorrelation function for \( m \) vectors of \( n \) elements each. Using the novel algorithm proposed in this paper, the elapsed calculation time goes as \( mxn^2 \), therefore it is less computer time consuming. Moreover, the novel procedure provides, together with the average speckle size, the error in speckle size calculation, as the standard deviation of the set of profile speckle sizes, as defined in Section 2.

The algorithm was tested first on calculated, generated speckled images, having different speckle profiles, placed both in rows and columns and randomly distributed and overlapped and it proved to provide a reasonable measure of the speckle size.

Once the test was successfully passed, the algorithm was tested on images taken from the real world, that is far field speckles recorded for milk fat particles in aqueous suspension at various dilutions and the variation of the speckle size with the SC concentration was recorded and presented in Section 3. The variation was found to be similar with the results reported in the literature for the same type of SCs and for calibrated latex balls.

The faster algorithm that was developed and tested can be used further on in developing an online application of monitoring the particle concentration in the very small range, that is ppm to hundreds of ppm.
REFERENCES


