A multiple exposure laser speckle contrast imaging (MELSCI) setup for visualizing blood perfusion was developed using a field programmable gate array (FPGA), connected to a 1000 frames per second (fps) 1-megapixel camera sensor. Multiple exposure time images at 1, 2, 4, 8, 16, 32 and 64 milliseconds were calculated by cumulative summation of 64 consecutive snapshot images. The local contrast was calculated for all exposure times using regions of 4 × 4 pixels. Averaging of multiple contrast images from the 64-millisecond acquisition was done to improve the signal-to-noise ratio. The results show that with an effective implementation of the algorithm on an FPGA, contrast images at all exposure times can be calculated in only 28 milliseconds. The algorithm was applied to data recorded during a 5 minutes finger occlusion. Expected contrast changes were found during occlusion and the following hyperemia in the occluded finger, while unprovoked fingers showed constant contrast during the experiment. The developed setup is capable of massive data processing on an FPGA that enables processing of MELSCI data in 15.6 fps (1000/64 milliseconds). It also leads to improved frame rates, enhanced image quality and enables the calculation of improved microcirculatory perfusion estimates compared to single exposure time systems.

**KEYWORDS**

blood flow, blood perfusion, FPGA, LASCA, LSCI, microcirculation, multiexposure

---

**INTRODUCTION**

Laser speckle contrast imaging (LSCI) is a noninvasive imaging technique for measuring blood perfusion. When coherent light is backscattered from an object an interference pattern, a speckle pattern, is formed on the imaging devices, such as a Complementary Metal Oxide Semiconductor (CMOS) or Charge Coupled Device (CCD) sensor [1]. The spatial and temporal properties of speckles were first researched in the 1960s, and later adopted for medical use in 1975 [2]. The principle is based on speckle fluctuations or movements that occur when the backscattered light is Doppler shifted by...
moving red blood cells in the tissue. Depending on the size of the Doppler shifts, blurring will occur as the speckles move during the exposure of the image sensor. The longer the exposure time of the image, the more the speckles move during each frame, and thus the blurrier the images get. This fact is used in LSCI to relate speckle images to the movement [3, 4]. The analysis method is also commonly referred to as laser speckle contrast analysis (LASCA).

A major drawback with LSCI is its inability to separate flow velocity (directional) or speed (nondirectional) from the amount of moving red blood cells when using single exposure images. It has been shown that LSCI is less sensitive to speed changes than laser Doppler flowmetry (LDF), whereas it is affected more by changes in optical properties of blood and the surrounding tissue, as compared to LDF [5]. To improve this, setups for multiple exposure times LSCI (MELSCI) have been proposed [3, 6, 7]. However, in order to capture images with several exposure times, either a reconfiguration of the camera exposure time or a time modulation of the laser was necessary in previous setups. This results in a slow acquisition process.

To overcome the slow acquisition process using a standard MELSCI setup Dragojević et al [8] used a high speed single-photon avalanche diode (SPAD) in order to capture low-resolution images with very short exposure time and inter-frame delay. These images were then accumulated numerically in the post processing to create longer synthetic exposure times. One limitation with the SPAD was the very low resolution (64 × 32 pixels). This problem was addressed by Sun et al [9, 10], who used a high-speed CMOS sensor coupled with a field programmable gate array (FPGA). Synthetic exposure times were created using the same technique as with the SPAD, in order to compare MELSCI and LDF. However, because of the high frame rate required to perform LDF (11.6 kHz in this case), it was not possible to expose and read out the whole 1.31-megapixel image at the same time. Instead, sequential subwindowing of the sensor was used to attain full-size images. This has the drawback that not all parts of the image are exposed at the same time. The highest presented frame rate in Sun’s work (at 1.31 megapixel) was 0.225 frames per second (fps), or one MELSCI dataset every fifth second. Therefore, it is not possible to follow fast dynamic changes with their technique.

The aim of this study was to develop a real-time FPGA implementation capable of processing high resolution MELSCI at high frame rates. The implementation includes data buffering and sorting, accumulation of consecutive images and parallel contrast calculations for an efficient work flow. The capability of the system to produce real-time visual feedback is demonstrated using a 1000 fps 1-megapixel CMOS camera during occlusion tests of real tissue.

2 MATERIALS AND METHODS

2.1 Multiexposure laser speckle contrast imaging algorithm

Multiple synthetic exposure times were obtained by accumulating images with a single, fixed exposure time in order to simulate longer integration times of the camera. The camera was configured to use an exposure time of 980 microseconds, and a frame rate of 1000 fps (1 millisecond/image). This gave an interframe time of 20 microseconds, which is short enough to consider two adjacent images to be continuous, as typical speckle decorrelation times are in the order of milliseconds. [1] These settings limited the camera resolution to 1024 × 1000 pixels. For convenience, the images with 980-microsecond exposure time in the rest of the paper will be referred to as having an exposure time of 1 millisecond.

The algorithm performed the accumulation of images, and the traditional variance and average calculations for nonoverlapping pixel submatrices [4, 11]. The submatrices were selected to be 4 × 4 pixels large, as a “power of 2”-size made it possible to better utilize the FPGA resources and improve the overall performance. Improved contrast algorithms have been proposed [12], which speeds up the processing time and might have enabled overlapping submatrices with good performance, but they were not used in this study. Figure 1 shows an overview of the algorithm from raw data to multiexposure contrast images.

![FIGURE 1](image-url) Overview of the algorithm from raw data to multiexposure contrast images. Capital T denotes exposure time of the images, while lowercase t denotes actual time.
The accumulation of exposure times was performed as a binary tree structure. A set of 64 images was captured and divided into pairs of subsequent images. For each of the pairs, the corresponding pixels in both images were added together, creating a single image with double the exposure time of the originals. This process was then iteratively repeated with the new images as the input, resulting in 64 images with an exposure time of 1 millisecond, 32 images with an exposure time of 2 milliseconds, ..., 2 images with an exposure time of 32 milliseconds and a single image with an exposure time of 64 milliseconds. In total, this produced 127 images with 7 different exposure times. For each of these, the local variance and intensity images were then calculated for each exposure time, maximizing the utilization of intensity images calculated using Eqs. (3) and (4) respectively, is shown in the right blocks denoted 1, 2 and 64 milliseconds are speckle images, and the blocks denoted \( \sigma^2 \) and \( I \) are the variance and intensity images, calculated with Eqs. (1) and (2). The mean of the variance images and of the average intensity images, calculated using Eqs. (3) and (4) respectively, is shown in the right.

![Diagram of binary tree summation and averaging algorithm.](image)

The mean variance image for exposure time \( T \) was calculated as:

\[
\sigma^2(T) = \frac{1}{16} \sum_{i=1}^{16} (x_i(T))^2 - \left( \frac{1}{16} \sum_{i=1}^{16} x_i(T) \right)^2
\]

(1)

and the local average intensity as:

\[
\langle I(T) \rangle = \frac{1}{16} \sum_{i=1}^{16} x_i(T),
\]

(2)

where \( x_i \) is the intensity of pixel \( i \) in the submatrix and \( T \) is the exposure time. The mean variance and mean average intensity images were then calculated for each exposure time. This ensured that all of the 64 original images were used for each exposure time, maximizing the utilization of the available information. This also reduced stochastic noise in the images. The mean variance image for exposure time \( T \) was calculated as:

\[
\langle \sigma^2(T) \rangle = \frac{1}{N} \sum_{n=1}^{N} \sigma^2_n(T)
\]

(3)

and the mean average intensity image as:

\[
\langle \langle I(T) \rangle \rangle = \frac{1}{N} \sum_{n=1}^{N} \langle I(T) \rangle_n,
\]

(4)

where \( N \) is the number of images for exposure time \( T \) (64 for 1 millisecond, 32 for 2 milliseconds, etc.). The principles are illustrated in Figure 2. From the 7 variance and 7 average intensity images, the contrast-square image for each exposure time was calculated as [1]:

\[
K_{\text{raw}}^2(T) = \frac{\langle \sigma^2(T) \rangle}{\langle \langle I(T) \rangle \rangle^2},
\]

(5)

where \( \text{raw} \) indicates that this contrast is not yet calibrated. Because the local contrast was calculated on nonoverlapping pixel submatrices the resolution of the contrast images was decreased by a factor 4 both horizontally and vertically, thus resulting in a resolution of 256 × 250 pixels. This is also illustrated in Figure 1.

It is worth noting that \( \langle \langle I(T) \rangle \rangle \) for different \( T \) actually contains the same information, only scaled differently because of \( T \). Therefore, the average intensity only has to be calculated for one of the 7 exposure times. The average for the other 6 exposure times can be easily derived from that one with a simple scale factor. The value that was actually calculated in the implemented system was \( \langle \langle I(64) \rangle \rangle \).

### 2.2 | Detector noise reduction

The variance and average intensity were described using the following model, similar to the one proposed by Valdes et al [13]:

\[
\sigma^2_{\text{measure}}(T) = \sigma^2_{\text{speckle}}(T) + \sigma^2_{\text{dark}}(T) + \sigma^2_{\text{noise}}(T)
\]

\[
\langle I(T) \rangle_{\text{measure}} = \langle I(T) \rangle_{\text{speckle}} + \langle I(T) \rangle_{\text{dark}} + \langle I(T) \rangle_{\text{ambient}}
\]

(6)

where \( \sigma^2_{\text{dark}}(T) \) and \( \langle I(T) \rangle_{\text{dark}} \) are the variance and intensity of the dark currents in the camera. These dark currents are always present, regardless of the illumination of the sensor, and will contribute to both the contrast and the intensity. The dark variance and intensity were measured by capturing 60 consecutive series of 64 images each, with the camera lens covered. The variance and intensity for each series were then averaged, and the averages were subtracted from the real measurements before the contrast was calculated using Eq. (5).
The intensity of the ambient light in the room, $(I(T)_{\text{ambient}})$, was eliminated from the model by performing all measurements in a dark room where the ambient light was negligible compared to the intensity of the laser.

The value $\sigma^2_{\text{noise}}(T)$ is an intensity-dependent variance related to shot noise [6]. It was measured by fitting a first degree polynomial to measurements of a fast rotating paper, illuminated with a white light source at a continuous range of distances. It was discovered that $\sigma^2_{\text{noise}}(T)$ was significantly different for each pixel for the camera used, and thus a different polynomial had to be fitted for each pixel. These polynomials were then used to interpolate the value of $\sigma^2_{\text{noise}}(T)$ for each pixel of the actual measurements, so that it could be subtracted before calculating the contrast values.

2.3 Calibration

The contrast values obtained from the algorithm were rescaled to fit the correct range from 0 to 1 by calibrating the system with measurements of the maximum contrast that could be measured for each exposure time, $K^2_{\text{max}}(T)$. This was obtained from measurements on a laser-illuminated stationary white paper, and the final contrast was then calculated as:

$$K^2(T) = \frac{K^2_{\text{raw}}(T)}{K^2_{\text{max}}(T)} \quad (7)$$

The minimum contrast $K^2_{\text{min}}(T)$ was also obtained, by measuring a laser-illuminated rapidly rotating paper. This should ideally be zero after the noise reduction, which is very close to what we observed. This is consistent with previous works [3], and $K^2_{\text{max}}(T)$ corresponds to the $\beta$-value presented in these. Both the $K^2_{\text{max}}(T)$ and $K^2_{\text{min}}(T)$ curves are shown in Figure 3. It is worth noting that $K^2_{\text{max}}(T)$ is essentially unaffected by exposure time.

2.4 Implementation on an FPGA and optical setup

We designed a system to perform the above algorithm in real-time. The setup is shown in Figure 4, with the most important parts numbered. High data throughput and low-latency computations were required for the real-time processing, which was solved by using a Kintex7 FPGA on a KC705 development board (1) (Xilinx, San Jose, CA, USA). The FPGA was connected to an EoSens 3CXP high speed camera (2) (Mikrotron, Unterschleissheim, Germany) via a CoaXPress cable, using an FPGA Mezzanine Card for CoaXpress (3) (Kaya Instruments, Nesher, Israel). The camera captured 1-megapixel images at 1000 fps, using 8 bit precision and a single color channel. The focal length of the camera lens was 12.5 mm and the $f$-number was set to 1.4. To connect the FPGA to the computer, a Gigabit Ethernet cable was used for transferring data, and two separate USB cables for configuration and command interface, respectively. A 780 nm single longitudinal-mode laser (4) equipped with an optical diffusor was mounted on the camera. The camera was placed approximately 200 mm from the imaged object, resulting in an imaged area of approximately $130 \times 130$ mm.

For real-time signal processing, the system should be able to process images faster than the camera could capture them, that is, one set of 64 images had to be processed in less than 64 milliseconds. This was achieved by using a pipelined design, in which different subsystems could work in parallel on different datasets. Data was sent between the subsystems using the peripheral Synchronous Dynamic Random Access Memory (SDRAM) and the smaller on-chip RAMs. An overview of the system design can be seen in Figure 5. A MicroBlaze Central Processing Unit (CPU) (Xilinx) was used as the control unit and user interface on the FPGA, communicating with the computer via Universal Asynchronous Receiver/Transmitter.

Images captured by the camera were received by the camera controller, which continuously wrote the data into the SDRAM. When 64 images had been written, a control signal was sent to the Kernel subsystem, which then read the data from the SDRAM in order to process it. In the Kernel subsystem the images were divided into submatrices and sorted into 8 RAMs. The sorting was done to have quick and easy access to corresponding submatrices in the 64 images. After the sorting, 8 parallel computation units performed the above algorithm on 1 RAM each, Eqs. (1)-(4), sharing the computational load in order to meet the real-time requirements of the system. The output from the Kernel subsystem was written back into the SDRAM, and a control signal was sent to the MicroBlaze, which initiated a Transmission Control Protocol (TCP) transfer of the results to the computer. The remaining steps of the algorithm, Eqs. (5)-(7), were performed on the computer in order to minimize numerical errors due to the integer mathematics in the FPGA. Note that this was still performed in real-time, in parallel with the computations on the FPGA, as the data

![Figure 3](image-url)
amount was reduced 25 times in the steps performed on the FPGA.

In order to speed up the system further, all memories (peripheral SDRAM and on-chip RAMs) were double buffered. By using two buffers in all memories, one subsystem could write to one buffer while another subsystem could read from the other buffer, minimizing idle time for all subsystems. This doubled the amount of required memory, but greatly increased performance. Once a buffer was processed, the active read and write buffers were switched, the new data were processed and the old data were overwritten. This enabled continuous readout from the camera, by making it possible for the camera controller to write raw camera data to one buffer in the SDRAM, while the Kernel worked on the data in the other buffer. The double buffering was also an essential part of the sorting algorithm, and allowed the system to process images faster than the camera could capture them.

2.5 In vivo measurement

The MELSCI system was utilized in an arterial occlusion and release experiment. A healthy 24-year-old Caucasian male, refraining from coffee and other substances affecting the microcirculation during the same day, was acclimatized
for more than 15 minutes in sitting position in a room with a temperature of 23°C. Measurements were done before, during and after a stimulus–response provocation where a finger was occluded using a small blood pressure cuff inflated to 200 mmHg for 5 minutes. A black low-reflecting background was used in the measurements. The measurement protocol was approved by the regional ethical review board at Linköping University, Linköping, Sweden (D.nr 2015/392-31).

3 | RESULTS

3.1 | Performance of the system/algorithm

The performance and utilization of the system are presented in Table 1. The time required to process one set of 64 images in the FPGA was 28 milliseconds. As the capture of one such set by definition took 64 milliseconds, the FPGA/camera system was capable of continuous acquisition and processing of data. By capturing 1000 fps, with each set of 64 images giving one set of multiexposure contrast images, the number of produced contrast-frames per second was

\[
\text{Frame rate} = \frac{1000}{64} = 15.625 \text{ frames/second.} \tag{8}
\]

The effective frame rate on the computer was much lower than the frame rate achieved by the FPGA system. This was due to the TCP transfer between the FPGA and the computer, which was significantly slowed down by the software library for the TCP stack, Light-weight Internet Protocol (LwIP). The reason for this bottleneck is probably due to LwIP running on a soft microprocessor, which shared resources with the rest of the processing system.

For describing the FPGA utilization the flip flops, look up tables (LUT) and on-chip block RAMs (BRAM) were, in our opinion, the most important resources for this particular system. The utilization and performance of the system are presented in Table 1.

4 | IN VIVO MEASUREMENT

Baseline contrast images for the different exposure times, taken before the occlusion, are presented in Figure 6. It can be seen that contrast decays with increasing exposure time, as expected. Contrast images taken after 5 minutes of occlusion are presented in Figure 7. The contrast increased in the occluded finger indicating a lowered blood flow. Contrast images taken immediately after the release of the occlusion pressure are presented in Figure 8. The hyperemia causes finger contrast to decrease to values well below baseline values.

An intensity-threshold was applied to the images in Figures 6–8. Any contrast-pixels where the average intensity was too low were masked, in order to only show reliable contrast values on the hand. The images have a size of 256 × 250 pixels, and depict an area of 130 × 130 mm.

A 10 × 10 pixel region of interest (ROI), marked in Figure 8 (64 milliseconds image), was selected on the provoked finger and on a control finger. The average contrast in the ROI on the provoked finger displayed a clear difference between the three time points depicted in Figures 6–8 (ie baseline, occlusion and reperfusion), using the same ROI for all images (Figure 9A). The corresponding contrast in the control finger ROI did not display any relevant difference between the 3 time points (Figure 9B).

5 | DISCUSSION

The main purpose of this study was to show the efficiency and applicability of the synthetic MELSCI algorithm and the use of an FPGA in order to provide real-time contrast images. The system implemented on the FPGA was capable of processing 64 1-megapixel images in 28 milliseconds. Hence, the signal processing can be done with continuous data acquisition without losing any frames.

Synthetic MELSCI implemented on an FPGA was utilized by Sun et al [9, 10]. They recorded 1024 320 × 320 pixel images at 15 kHz (exposure time 66.6 microseconds) for comparing Laser Doppler Imaging (LDI) and LSCI, or a set of subframes of 1280 × 32 pixels at 11.6 kHz (exposure time 85 microseconds). Unlike our solution, these approaches do not allow for a continuous acquisition of high resolution (1 megapixel) images due to the choice of a short exposure time.

The transfer speed of the Gigabit Ethernet connection that was used should theoretically be fast enough. However, complications with the software library for the TCP/IP stack, LwIP, slowed the transfer speed to less than a 10th of the theoretical maximum. This data transfer problem is very similar to what Sun et al found in their design [10]. It is possible that these bottlenecks could be reduced by improving the existing interfaces and software libraries used in the transfer between the FPGA and the computer, or by simply using a more powerful FPGA. However, it is much more likely that the solution is to move from a pure FPGA solution, to a system-on-chip containing both an FPGA and a hard CPU optimized for the task of high bandwidth data

| TABLE 1 | Performance metrics of the system and utilization of FPGA resources |
|--------------------------|--------------------------|
| FPGA system performance | Frame rate of the system |
|                         | 15.6 fps |
| FPGA utilization        | Flip flop utilization |
|                         | 23%     |
|                         | LUT utilization |
|                         | 39%     |
|                         | BRAM utilization |
|                         | 62%     |
| Complete system performance | Transfer speed from FPGA to computer |
|                         | 49 Mb/s |
| Effective frame rate on computer | ~1.5 fps |
transfer, unlike the soft CPU implemented in our design. As the transfer speed itself was not the focus of this work, we decided to leave these improvements for the next version of the system.

When examining the contrast curves in Figure 9, the smooth shape between adjacent exposure times is apparent. This smooth shape remains for individual pixels without averaging over a ROI and is a result from a high correlation between exposure times. The high correlation is natural since the contrast for all exposure times is calculated from the same measured data, for example, the contrast for the 1-millisecond exposure time is an average of 64 different 1-millisecond contrast images. This is an important difference from previous multiple exposure systems, for example, Parthasarathy...
et al [6] and Thompson et al [14], where the contrasts from different exposure times are recorded at different time instances with a large inter-frame delay. Sampling the contrast from different realizations of the same stochastic process in that way results in noisy contrast decay. A smooth contrast decay is important to further analyze the MELSCI data.

Stochastic noise originating from the limited number of realizations of the speckle pattern, that is a stochastic process, remains in our data. This noise is manifested by a small varying offset and tilt in the contrast decays. In order to eliminate that noise further, averaging over consecutive MELSCI data sets is the only way to go. It should also be noted that since we utilize all data to a maximal extent by reusing the same sampled data for all exposure times, all noise is suppressed maximally by averaging.

The results of the in vivo measurements clearly show that the effect of the occlusion of the finger as well as the post occlusive reactive hyperemia affect the contrast for all exposure times. It is also clear, especially when examining Figure 9, that the effects are different for the different exposure times. It can thus be concluded that the different exposure times contain (partly) different information about the actual tissue perfusion. It has for example previously been shown that short exposure times are more sensitive to blood speed changes while longer exposure times are mainly sensitive to blood amount changes [5]. It has also been concluded in previous studies that multiple exposure times are necessary to retrieve perfusion data that better reflects the actual perfusion in the sampling volume and, potentially, reveal the speed distribution of blood [3, 5, 14, 15]. Algorithms utilizing the multiple exposure times can be implemented in this system, and the system can then be used to evaluate the applicability and robustness of such algorithms on real data.

**FIGURE 8** Contrast ($K^2(T)$, Eq. (7)) images of a hand during reperfusion in a finger, for 7 different synthetic exposure times. In the 64-millisecond image, a region of interest is marked on the provoked finger (upper left square) and a control finger (lower right square).

**FIGURE 9** Contrast curves for the images in Figures 6–8, in a region of interest selected on the provoked finger (A) and a control finger (B).
In practice, implementing contrast calculations on an FPGA as done in this study can lead to two important improvements compared to commercial LSCI systems available today. First, because data reduction in form of contrast calculations (16-to-1 pixels in our implementation) and averaging can take place on the FPGA, the communication interface to for example a computer will not be a limiting factor anymore. This will result in better image quality and/or higher frame rates and/or higher resolution perfusion images. Secondly, an improvement with much higher potential is that the multiple exposure times will enable the calculation of perfusion estimates superior in accuracy and predictability compared to the perfusion estimate that can be calculated from single exposure time systems. It may even lead to quantitative and speed resolved perfusion estimates as has been demonstrated for LDF [16]. In that case, fundamentally new ways to examine the microcirculation with a combined high spatial and temporal resolution will be a reality, with potential improved diagnosis and treatment of people suffering from diseases causing impaired microvascular function, such as diabetes.

ACKNOWLEDGMENTS
The authors would like to thank Andreas Ehliar (PhD), Christopher Hallberg (MScEng) and Alfred Zickerman Bexell (MScEng) for their contribution to this project. This study was financially supported by the Swedish Research Council (grant no. 2014-6141) and by the CENIIT research organization within Linköping University (project id. 11.02).

CONFLICT OF INTEREST
The authors have no conflict of interest.

REFERENCES
[1] D. A. Boas, A. K. Dunn, J. Biomed. Opt. 2010, 15, 011109.
[2] J. D. Briers, Opt. Appl. 2007, 37, 139.
[3] D. Zölei-Szénási, S. Czimmer, T. Smausz, F. Domoki, B. Hopp, L. Kemeny, F. Bari, I. Ivanyi, J. Eur. Opt. Soc.-Rapid. 2015, 10, 15033.
[4] J. D. Briers, S. Webster, J. Biomed. Opt. 1996, 1, 174.
[5] I. Fredriksson, M. Larsson, J. Biomed. Opt. 2016, 21, 126018.
[6] A. B. Parthasarathy, W. J. Tom, A. Gopal, X. Zhang, A. K. Dunn, Opt. Express 2008, 16, 1975.
[7] A. B. Parthasarathy, S. M. Shams Kazmi, A. K. Dunn, Biomed. Opt. Express 2010, 1, 246.
[8] T. Dragojević, D. Bronzi, H. M. Varma, C. P. Valdes, C. Castellvi, F. Villa, A. Tosi, C. Justicia, F. Zappa, T. Durduran, Biomed. Opt. Express 2015, 6, 2865.
[9] S. Sun, B. R. Hayes-Gill, D. He, Y. Zhu, S. P. Morgan, Opt. Lett. 2015, 40, 4587.
[10] S. Sun, B. R. Hayes-Gill, D. He, Y. Zhu, N. T. Huynh, S. P. Morgan, Opt. Lasers Eng. 2016, 83, 1.
[11] T. M. Le, J. S. Paul, H. Al-Nashash, A. Tan, A. R. Luft, F. S. Sheu, S. H. Ong, IEEE Trans. Med. Imaging 2007, 26, 833.
[12] W. J. Tom, A. Ponticorvo, A. K. Dunn, IEEE Trans. Med. Imaging 2008, 27, 1728.
[13] C. P. Valdes, H. M. Varma, A. K. Kristoffersen, T. Dragojević, J. P. Culver, T. Durduran, Biomed. Opt. Express 2014, 5, 2769.
[14] O. B. Thompson, M. K. Andrews, J. Biomed. Opt. 2010, 15, 027015.
[15] D. Briers, D. D. Duncan, E. Hirst, S. J. Kirkpatrick, M. Larsson, W. Steenbergen, T. Stromberg, O. B. Thompson, J. Biomed. Opt. 2013, 18, 066018.
[16] I. Fredriksson, M. Larsson, T. Strömberg, J. Biomed. Opt. 2010, 15, 057002.

How to cite this article: Hultman M, Fredriksson I, Larsson M, Alvandpour A, Strömberg T. A 15.6 frames per second 1-megapixel multiple exposure laser speckle contrast imaging setup. J. Biophotonics. 2018;11: e20170069. https://doi.org/10.1002/jbio.20170069