CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Sub-pixel point detection algorithm for point tracking with low-power wearable camera systems: a simplified linear interpolation
Authors	Wilk, Mariusz P.;Urru, Andrea;Tedesco, Salvatore;O'Flynn, Brendan
Publication date	2017-06-01
Original Citation	Wilk, M. P., Urru, A., Tedesco, S. and O'Flynn, B. (2017) 'Sub-pixel point detection algorithm for point tracking with low-power wearable camera systems: a simplified linear interpolation', 28th Irish Signals and Systems Conference (ISSC), Killarney, Ireland, 20-21 June. doi:10.1109/ISSC.2017.7983629
Type of publication	Conference item
Link to publisher's version	http://www.issc.ie/site/view/7/ - 10.1109/ISSC.2017.7983629
Rights	© 2017, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-11-03 15:49:05
Item downloaded from	https://hdl.handle.net/10468/5450

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Sub-Pixel Point Detection Algorithm for Point Tracking with Low-Power Wearable Camera Systems

A Simplified Linear Interpolation

Mariusz P. Wilk, Andrea Urru, Salvatore Tedesco, Brendan O'Flynn
Tyndall National Institute
University College Cork
Cork, Ireland
mariusz.wilk@tyndall.ie

***Abstract*— With the continuous developments in vision sensor technology, highly miniaturized low-power and wearable vision sensing is becoming a reality. Several wearable vision applications exist which involve point tracking. The ability to efficiently detect points at a sub-pixel level can be beneficial, as the accuracy of point detection is no longer limited to the resolution of the vision sensor. In this work, we propose a novel Simplified Linear Interpolation (SLI) algorithm that achieves high computational efficiency, which outperforms existing algorithms in terms of the accuracy under certain conditions. We present the principles underlying our algorithm and evaluate it in a series of test scenarios. Its performance is finally compared to similar algorithms currently available in the literature.**

Keywords—sub-pixel; point detection; low power; wearable; vision; approximation

I. INTRODUCTION

Visual point detection is an important task in the field of digital image processing. The ability to accurately and precisely estimate the position of a given point of interest is fundamental in point tracking as well as many other image processing applications, including: object detection (template matching), pattern recognition, etc. The success of such applications in meeting their objectives relies on the performance of the underlying lower-level algorithms, such as image segmentation and feature detection, of which point detection is an integral part [1]. A variety of interesting application spaces emerge for visual point tracking with the continuous developments in sensor technology. Some of the recent advances in lens-less vision sensor technologies show that the dependency on traditional lenses, often the largest component of a typical vision sensor system, can be eliminated, thus significantly reducing the physical size while achieving an equivalent performance level [2, 3]. Although such camera systems do not perform to the same optical level as traditional cameras do, these can still be suitable for applications that involve point detection and tracking. This can be particularly significant in the context of low-power, miniaturized and connected wearable motion tracking devices.

Low-power wearable vision systems, however, face several challenges. Digital image processing, for example, can be computationally intensive. Whereas it is not a limiting factor in traditional image processing applications that utilize virtually unlimited resources, as far as the application designer is concerned, it can be very significant if the processing is carried out on miniaturized low-power wearable platforms.

There are several factors that contribute to the computational complexity of an algorithm. One of the most challenging situations is the fact that many applications require the system to process the image frames at interactive rates, e.g. 25 frames pes second or more. This can be extremely challenging when tens of frames must be processed each second. Moreover, it is often the case that image processing algorithms comprise multiple stages, therefore a given input frame must be processed more than once before proceeding to the following frame. Furthermore, the resolution of the imaging sensor has a major impact on image processing speed. Although a higher resolution helps capture more information from the environment, it occurs at the expense of either increasing the processing power of the hardware or decreasing the frame rate of the output. On the other hand, lower resolution image frames can help to increase the frame rate, but the accuracy and precision of the output are often compromised.

One of the possible ways to resolve these challenges can be to assume a semi-controlled ambient environment to eliminate unnecessary sources of noise. In a typical point tracking application, the point detection algorithm is focused on finding the coordinates of blobs that represent the points in the image. The blobs can be extracted from an image by assuming that the points to be tracked are specific sources of light, e.g. infrared LEDs, and the vision sensor is fitted with an appropriate optical filter. This net result being that only the sources of light that represent the points are captured by the sensor. Thus, the noise floor in the image should be low and uniform, and the magnitude of the blobs' peaks should be well above the noise floor, making them easily identifiable. The intensity of the sources of light can be controlled in such a way that no pixels in the sensor are saturated. Moreover, the Field-of-View (FOV)

of the imaging sensor can be reduced to pixels that lie within such a radius that the geometric distortions can be neglected [4]. Under these conditions, the sources of light should appear as Point Spread Functions (PSF) with Gaussian characteristics [5], over an area between 3x3 and 6x6 pixels. Coupling the sensor with the ambient environment can significantly increase the efficiency of point detection algorithms at pixel level. Indeed, the pixel-level point finding algorithms can be limited to finding the local maxima in the image. Secondly, the resolution of the sensor can be decreased to reduce the number of pixels to be processed in each frame, thus further increasing the speed of pixel-level point detection algorithms.

However, in most point tracking applications, a lower resolution image decreases the accuracy of point detection to an unacceptable level at the pixel level. This limitation can be overcome by finding the coordinates of the points at sub-pixel level. The true coordinates of the points are located about the detected peaks at the pixel level. The coordinates of the points can be refined to sub-pixel level by inspecting the neighbourhood of the peak pixel intensity, thus overcoming the limitations of the pixel resolution of the imaging sensor. The neighbouring pixel intensities contain the necessary information required to help estimate the location of the true intensity peak at the sub-pixel level. Fig. 1 depicts a typical pixel-level and sub-pixel level intensity peak sampled along the x-dimension. The super-resolution methods for sub-pixel point detection are well documented in literature [5, 6]. Historically, the ratio of the time taken by a computer program to detect a point at pixel level was much higher than the time taken to detect the point at sub-pixel level. Therefore, more attention was usually paid to the accuracy of the sub-pixel detection algorithms than the time requirements of the computation as this was seen to be negligible due to high processing power of the computing platform. This is not always the case in the context of ultra-low-power wearable platforms. Such resource constrained systems by their nature have limited resources. This is particularly the case for those that rely on the intelligent coupling of the sensor with the ambient environment as described above. In this case, the point detection at pixel level can be simplified to such a degree that the timing of a given sub-pixel detection algorithm may become as important as its accuracy; thus, this work considers these two criteria as equally important.

The goal of this work is to compare and contrast the state-of-the-art algorithms in this field to a proposed novel approach that could both accurately and efficiently estimate the peak intensities at sub-pixel level. This paper is structured around three main stages. Firstly, the Simplified Linear Interpolation (SLI) algorithm for sub-pixel point detection is proposed. Then its performance is evaluated and compared to two similar and well established algorithms in literature. Finally, the work is concluded with the analysis of the major findings and suggestions for future works.

II. Methodology

State-of-the-Art

Linear interpolation methods assume that a linear relationship exists between the points surrounding the interpolated value. It is one of the simpler and often most efficient ways to perform the interpolation, such as that based on the 1st order Newton's Divided Difference method [7]. However, it cannot be directly applied to sub-pixel peak detection. The identification of the sub-pixel point source is different from the typical problems that use linear interpolation. Whereas a typical interpolation problem involves finding the intensity value at a specific and known location, the sub-pixel peak detection is aimed at finding the coordinates of the true intensity peak, where neither the coordinates nor the intensity of the true peak are known. Therefore, the detection algorithms may only rely on the intensity values of the pixels that surround the true peak, as in Fig. 1. The coordinates of the intensity peak at sub-pixel level are defined by *x* and *y*, as in (1), where *X* and *Y* are the pixel-level x-y coordinates, and δ_x and δ_y represent the displacements, or sub-pixel offset, of the true peak from the detected pixel-level peak at the coordinates *X*-*Y*. Thus, the pixel-level point coordinates are refined to sub-pixel level by finding the values of δ_x and δ_y. Fig. 1 shows a typical 1-dimensional (1D) scenario with the Gaussian PSF sampled at the pixel resolution with the peak coordinate refined to the sub-pixel level.

$$x = X + \delta_x;\ y = Y + \delta_y \tag{1}$$

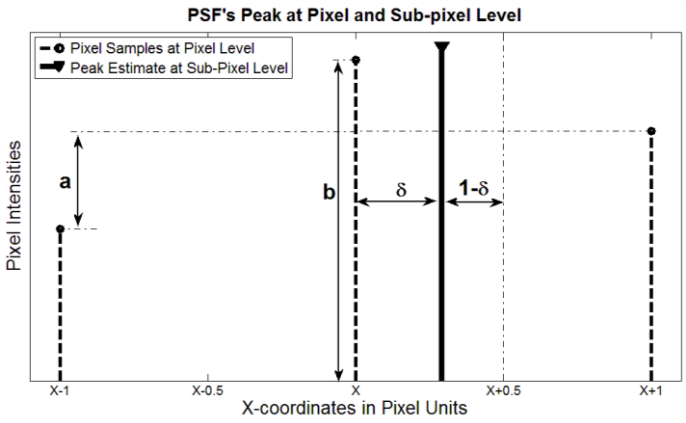


Figure 1: Typical 1D Gaussian PSF with peak at mean $\mu = X + \delta$ and std. dev. σ=1, sampled at pixel intervals: $X - 1,\ X,\ X + 1$

One of reference algorithms covered in the literature is the Linear Interpolation (LI), as described in [6]. It is computationally efficient when compared to other comparable algorithms. It leverages the assumption that the spread of pixel intensity values around the peak is defined by a linear relationship. Therefore, it defines δ_x as half the ratio of the difference between the preceding and the following pixel intensities (*a* in Fig. 1) to the difference between the peak pixel intensity and the lower peak of the two surrounding pixels (the peak located at X-1 in Fig. 1). Its accuracy is lower when compared to slower methods, such as the Gaussian

Approximation (GA), [6]. The GA, similarly to the LI, exploits the pixel around the observed intensity peak, but instead of assuming a linear relationship, it assumes a Gaussian spread of the intensities around the observed peak. It defines the sub-pixel offset δ_x in a similar way to that of the LI, but it differs in that it is based on a ratio of natural logarithms of the pixel intensities around the observed peak intensity.

There exists a range of other algorithms for super-resolution point detection. Some of these algorithms demonstrate very good accuracy and robustness in the presence of noise, but they are too complex from computational requirements' point of view. It makes them unsuitable in the context of the considered application space. For this reason, we decided to focus on the LI and GA. The LI was chosen, because it was mathematically the closest to the proposed SLI. The GA was chosen because it achieved the best performance in terms of accuracy.

SLI Algorithm

The proposed SLI algorithm achieves a faster calculation of the sub-pixel offset δ_x, compared to other methods. This section discusses the SLI algorithm in detail. The approach is based on linear interpolation but it uses the assumption of a linear relationship differently to methods described in the State-of-the-Art section. The underlying principles of the SLI algorithm can be explained using the trigonometric properties of similar triangles, as shown in Fig. 2. The pixel-level intensities of the peak and the two surrounding pixels, from Fig. 1, are approximated to the sides *a* and *b* of the two similar triangles. Similarly, the unknown sub-pixel offset from the observed pixel-level peak, δ, forms the horizontal side of the smaller triangle. The uncertainty area, i.e. the distance between *X* and *X±0.*5, as shown in Fig. 1, is equal to one, because this is the maximum absolute value that the sub-pixel offset δ_x may have around the given observed peak without having an error at pixel-level. Indeed, δ_x lies within ±0.5, as depicted in Fig. 1 and Fig. 2.

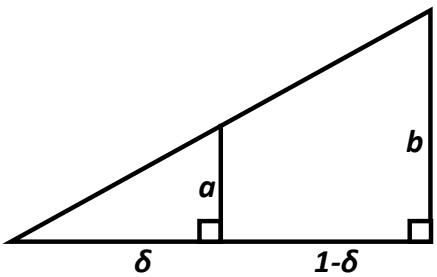


Figure 2: Pixel Intensity Approximation to Similar Triangles

The SLI relates the pixel intensities at and around the observed peak to the sub-pixel offset δ_x as a ratio of the difference between the pixel intensities of the two pixels surrounding the observed peak to the pixel intensity of the observed peak, as in (2):

$$\delta_x = \frac{a}{b} = \frac{f(X+1)-f(X-1)}{f(X)}; \quad \delta_x \in < -0.5, 0.5 > \tag{2}$$

The maximum value of the computed δ_x is capped to $\delta_x = \pm 0.5$ pixel. Moreover, due to the way the numerator of SLI is constructed, the sign of the resultant sub-pixel offset δ_x is determined intrinsically. It is clear that this approach can help – reduce the amount of required computations, thus increasing the speed of the execution, but it can likely compromise the mean accuracy of the measurement. There may exist such circumstances under which the SLI's performance may be comparable to that of the more complex algorithms, as described in sections *III-IV*.

III. Simulations

The performance of the SLI algorithm has been evaluated against the GA and LI algorithms using two criteria, i.e. the Root Mean Square Error (RMSE) and relative time of execution. The RMSE criterion was chosen to estimate the error over a large set of statistically random input parameters. The PSF was modelled using a 1D Gaussian distribution with generic centre. The two input parameters were the mean μ and standard deviation σ that vary randomly within specific intervals, as in (3):

$$\mu = X + \delta_\mu; \ \delta_\mu \in < 0, 0.5 >; \ \sigma \in < 0.5, 3 > \tag{3}$$

The choice of the centre of the observed peak at pixel level was arbitrary as it does not affect the results. The range of values of δ_μ was chosen to simulate every possible positive sub-pixel offset that may occur. The negative values of δ_μ were ignored as the results would have been the mirror of the results obtained with the positive δ_μ due to the assumption of the symmetry of the PSF. The range of the standard deviation values was selected to simulate the most likely variations in the spread of the PSF, detected by the vision sensor, typically caused by the changes in the properties of the ambient light sources, such as the changes in the intensity or the distance from the vision sensor.

The simulations were carried out as follows:

- For each pair of <μ, σ> input parameters, the related PSF model was generated depending on the scenario: both input parameters were random, one parameter was set to a specific value while the other was random;
- The pixel intensity values were sampled at the integer pixel locations around the observed peak;
- The sampled intensity samples were passed to the GA, LI and SLI algorithms to compute the sub-pixel offset δ_x;
- The error in the algorithms' output was computed, which is defined as the difference between δ_x and δ_μ, from (1) and (3), as in (4):

$$error = \delta_\mu - \delta_x \tag{4}$$

The Root Mean Square Error (RMSE) is then defined as in (5), where *N* is the number of iterations:

$$RMSE = \sqrt{\frac{\sum error^2}{N}};\ N = 10^6 \quad (5)$$

The value of *N* was chosen to be relatively large to establish the mean performance over a large set of input parameter pairs.

The relative time of execution was defined as the mean time taken by each algorithm to return the result, i.e. as the mean of the execution times t_i of each iteration i over all *N* iterations, (6):

$$Relative\ Mean\ Time = \frac{\sum_{i=1}^{N} t_i}{N} \quad (6)$$

IV. Results and Discussion

Each algorithms' performance was evaluated under three main scenarios:

1. Both the mean μ and std. dev. σ were random, and contained in intervals defined in (3);
2. Worst case scenario for std. dev. σ=3 and random mean μ, as defined in (3);
3. Worst case scenario for mean, i.e. $\delta_\mu = 0.5$ and random std. dev. σ, as defined in (3);

All results of the simulations are tabulated in Table 1. The SLI achieved the shortest relative mean execution time of all methods. It was the least accurate method of the three algorithms in all scenarios. The LI outperformed it by a factor of approximately three, given the mean performance of the Scenario 1. Although, the SLI achieved the shortest relative execution time in these three scenarios, the accuracy was significantly lower compared to the other methods.

It is worth noting that the RMSE of the GA was consistently low in all scenarios. This result is achieved for two reasons. Firstly, the GA assumes a Gaussian spread of pixel intensity values and the way the simulated PSF was modelled was very close to it. Secondly, we decided to not superimpose any white noise on the simulated PSF, which would have altered its performance [6]. Instead, the relative execution time of the GA was in the main focus and was used in the evaluation while the RMSE values were added to the table for consistency reasons. While the GA was the most accurate method under these scenario, it had the highest relative execution time; notably, its execution time was approximately twice as long as that of the SLI.

Table 1: Performance Evaluation Results

	SLI		**LI**		**GA**	
	RMSE	***Time*** $[10^{-6}]$	***RMSE***	***Time*** $[10^{-6}]$	***RMSE***	***Time*** $[10^{-6}]$
1	0.1462	0.717	0.0483	1.21	0.0010	1.69
2	0.2279	0.686	0.0589	1.17	0.0010	1.68
3	0.2433	0.716	0.0008	1.18	0.0010	1.67

A set of preliminary simulations revealed an interesting behaviour of the SLI around $\sigma = 1.2$. Therefore, an additional scenario was included in the evaluation process. The SLI demonstrated excellent performance in this scenario, wherein it outperformed the LI with respect to both selection criteria; the RMSE was 0.0026 and the relative time was 0.705. Results of the simulations are shown in Fig. 3 and Fig. 4. Fig. 3 demonstrates how the error in SLI's output varies as a function of the standard deviation σ for four fixed values of δ_μ; one with the extreme value of $\delta_\mu = 0.5$ for which the error is expected to be the largest and three others for which the error should be closer to the RMSE value. A marker in Fig. 3 is set at $\sigma = 1.2$ to emphasize the *error* for the different values of δ_μ. It is clear that the error is limited and that all four curves intersect at this value of σ. Fig. 4 further supports this observation by showing the relationship between the true δ_μ and the computed δ_x, over the full range of the sub-pixel offset values. This figure suggests that there is a linear relationship between δ_μ and δ_x at $\sigma = 1.2$; which means that the *error* is not only small but also uniform across the entire range of δ_μ.

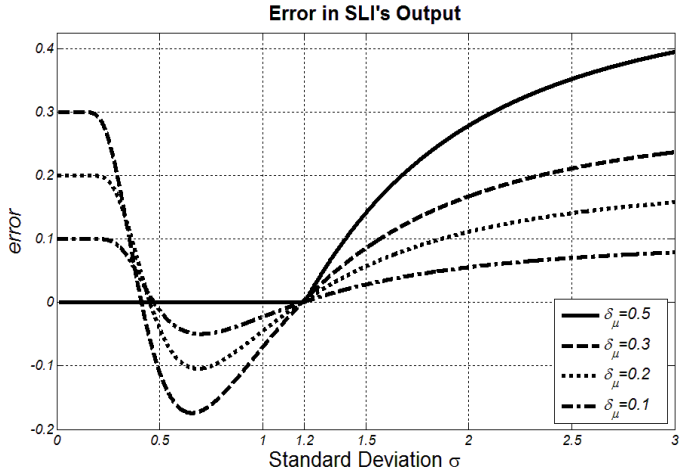


Figure 3: Additional Scenario: *error* for four values of δ_μ as a function of σ

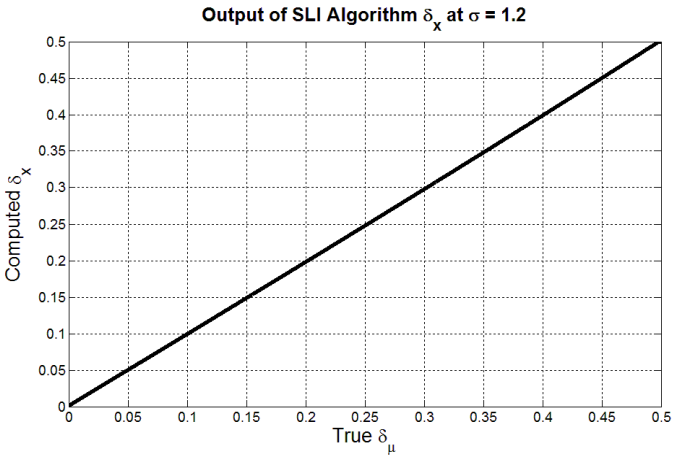


Figure 4: Additional Scenario: True δ_μ vs computed δ_x by the SLI at constant σ=1.2

The results of this additional scenario indicate that, despite the simplicity of SLI, this method presents an operating point, at which the *error* would be low and uniform. It may be possible to recreate such conditions (by coupling the vision system and

the ambient environment) so that this performance level may be achievable also with a real-world setup.

A mathematical explanation for such performance is suggested by the comparison of the denominators of the LI and SLI. These values can be approximated under specific circumstances. Thus, it could be considered a simplified approximation of the LI, as in (7):

$$2\big(f(X) - f(X-1)\big) \cong f(X) \tag{7}$$

Fig. 5 shows a simulated Gaussian distribution that reproduces these conditions on a specific example with $|\delta_\mu| = 0.5$, $2(f(X) - f(X-1)) \cong f(X) \cong 0.3048$. At this value of σ and δ_μ, the ratio (R) of $f(X-1)$ to $f(X)$ is equal to approximately a half, as shown in Fig. 5.

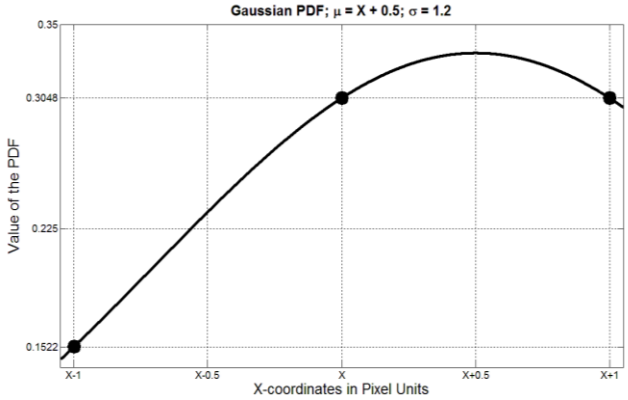


Figure 5: Peak of a Normal Distribution with sub-pixel offset $\delta_\mu = 0.5$ and $\sigma = 1.2$

Although, this relationship exists only at $|\delta_\mu| = 0.5$, it could be possible to apply it to the complete range of values of δ_μ over all values of standard deviation σ in the range. This possibility was investigated by modifying the denominator of SLI with a parameter α whose value depends on σ, as in (8)-(9):

$$2\big(f(X) - f(X-1)\big) = \alpha(\sigma) f(X) \tag{8}$$

$$\alpha(\sigma) = 2\left(1 - \frac{f(X-1)}{f(X)}\right) \tag{9}$$

An algorithm was developed to evaluate this approach (SLI_A). Initially, a look-up table was generated using a constant $\delta_\mu = 0.25$, for which the *error* was expected to be the largest. A set of Normal Distributions was generated for each value of σ, as in (3), except for the constant $\delta_\mu = 0.25$. The value of σ was estimated by computing the ratio R from the generated PDF, which was one of the most efficient methods. Therefore, for each value of σ, the ratio R and the parameter α were computed, quantized and saved in the look-up table. The resultant SLI_A algorithm had two additional stages, when compared to the original SLI: the computation of the ratio R and a search procedure through the quantization steps in the look-up table. Subsequently, the performance of the SLI_A was compared to the original version, SLI, using Scenario 1. The RMSE of the SLI_A was almost identical to that of the SLI, thus no decrease in RMSE to justify the increased computational complexity. The SLI_A was further investigated to try to estimate conditions under which it could out-perform the SLI; by assuming the value of α depended on both δ_μ and σ. The Scenario 3 was used to validate it, with the exception for the constant $\delta_\mu = 0.25$. These results are shown in Table 2; the asterisk implies Scenario 3 with $\delta_\mu = 0.25$. In this scenario, the SLI_A outperformed the SLI. Moreover, its RMSE approached that of the LI, shown in Table 1. However, while it did achieve low RMSE, it was impractical because α depends on the variable that the algorithm is aimed to compute. A more complex approach to solving this problem can be based on a recursive set-up which relies on past value(s) of δ_x to compute the current value of δ_x. However, this may be challenging, as, even though the best performance of the SLI_A could approach LI's accuracy, the computational complexity associated with it would be far greater, thus making it impractical.

Table 2: Performance Evaluation of SLI_A

	SLI_A		**SLI**	
	RMSE	***Time*** $[\mathbf{10^{-6}}]$	***RMSE***	***Time*** $[\mathbf{10^{-6}}]$
Scenario 1	0.1347	2.24	0.1464	0.674
Scenario 3*	0.0782	2.26	0.1318	0.627

V. Conclusions and Future Works

In this work, a novel sub-pixel point detection algorithm was proposed. It was derived to show how a Gaussian PSF could be approximated to a linear interpolation in a simplified way using the trigonometric principles of similar triangles, with the aim of achieving higher computational efficiency. The performance of the algorithm was evaluated and compared against two existing methods in a series of simulations. These three algorithms were subjected to different test scenarios. The performance criteria were the RMSE and relative execution time. The SLI was the fastest method in all cases. It underperformed the other two methods in the first 3 scenarios in terms of the RMSE. However, it yielded excellent results in the last scenario where it provided a better accuracy than the slower but generally more accurate LI algorithm.

This simulation comparison revealed that SLI can perform very well under specific conditions, i.e. when the standard deviation σ≈1.2. This could be a realistic scenario in a set-up where a wearable vision sensing platform is intelligently coupled with the ambient environment to ensure that the ambient conditions in vision sensor's FOV are close to those simulated ones. If these conditions are met, then an accurate point detection at sub-pixel level should be possible in the low-power wearable setup using the more computationally efficient SLI algorithm.

The future works will involve the validation of these simulations in an experimental setup that will resemble the assumed conditions in this work.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2006.

[2] P. Gill and T. Vogelsang, "Lensless Smart Sensors: Optical and thermal sensing for the Internet of Things," in *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, 2016, pp. 1-2.

[3] S. J. Byrnes, A. Lenef, F. Aieta, and F. Capasso, "Designing large, high-efficiency, high-numerical-aperture, transmissive meta-lenses for visible light," *Optics Express*, vol. 24, no. 5, pp. 5110-5124, Mar 2016.

[4] F. Heide, M. Rouf, M. B. Hullin, B. Labitzke, W. Heidrich, and A. Kolb, "High-quality computational imaging through simple lenses," *ACM Trans. Graph.*, vol. 32, no. 5, pp. 1-14, 2013.

[5] A. Small and S. Stahlheber, "Fluorophore localization algorithms for super-resolution microscopy," *Nat Meth*, Review vol. 11, no. 3, pp. 267-279, 03//print 2014.

[6] R. B. Fisher and D. K. Naidu, "A Comparison of Algorithms for Subpixel Peak Detection," in *Image Technology: Advances in Image Processing, Multimedia and Machine Vision*, J. L. C. Sanz, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 385-404.

[7] A. Kaw and E. E. Kalu, *Numerical Methods with Applications: Abridged*. Lulu.com, 2008.