Received: 00 Month 0000 | Revised: 00 Month 0000 | Accepted: 00 Month 0000

DOI: xxx/xxxx

ARTICLE TYPE

Bessel Image Sharpness Analysis, Advances in Bessel Image Sharpness Analysis

Arjuna P.H. Don*[1] | James F. Peters[1,2]

[1]Computational Intelligence Laboratory, University of Manitoba, WPG, MB, R3T 5V6, Canada

[2]Department of Mathematics, Faculty of Arts and Sciences, Adiyaman University, 02040, Adiyaman, Turkey

Correspondence
Email: pilippua@myumanitoba.ca

Summary

This paper introduces advances in the use of Bessel functions (cylinder functions) in achieving optimal sharpness of shapes delineated by collections of holes (dark valley regions) and bright regions in different areas of a video frame. An image hole is an island of low voxel intensities surrounded by varying voxel intensity peaks. The basic approach is to identify dominant collections of lights (high voxel intensity peaks) and darks (low voxel intensity clusters) in video frames. A main finding in this paper is that surface objects (recorded in images) are sharper wherever there are preponderant contrasting differences between image lights and darks. These contrasting differences lead to the highly accurate detection of shape holes (dark valleys) that delineate surface shapes recorded in sequences of video frames). With the use of cylindrical Bessel functions, contrasting differences between peaks and valleys can be measured in terms of voxel intensities and voxel indices.

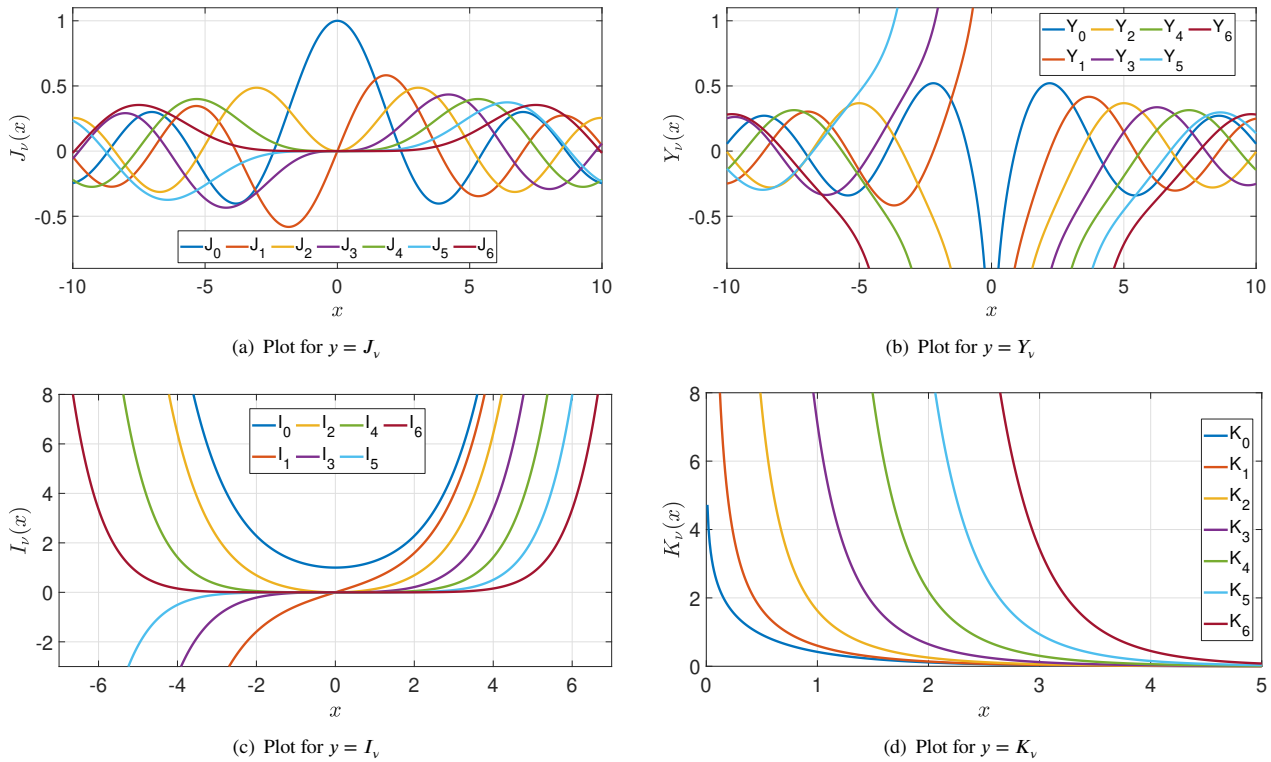
KEYWORDS:
MCS 2020 33C10 (Bessel and Airy functions, cylinder functions); 54H30 (Applications of general topology to computer science)), Sharpness metric, Betti numbers, Image assessment

1 | INTRODUCTION

This paper introduces a development in utilizing Bessel functions (cylinder functions) in identifying highly coherent collections of high contrasting peaks and valleys (holes) found in predominant surface shapes recorded in video frames. The end result of this research is the identification of optimal-nuclei at the centers of vortexes covering those image object shapes that are sharper than those of their neighbors. This work is part of ongoing research concerning the use of Bessel function in the detection of sharp patterns in images[1] and multi-zonal images[2,3].

2 | PRELIMINARIES

This section briefly explains the type of Bessel functions that are used in the paper and the correlation between image sharpness and contrast changes.



(a) Plot for $y = J_\nu$

(b) Plot for $y = Y_\nu$

(c) Plot for $y = I_\nu$

(d) Plot for $y = K_\nu$

FIGURE 1 Bessel functions for $\nu \in [0, 6]$

2.1 | Modified Bessel Differential Equation

2.1.1 | Bessel's Equation

The Bessel's equation is a second order ordinary differential equation of the form which often found when solving boundary value problems.

$$x^2 \frac{d^2y}{dx^2} + x\frac{dy}{dx} + \left(x^2 - \nu^2\right) y = 0 \tag{1}$$

These problems include, separable solutions to Laplaces equation or the Helmholtz equation, especially when working in cylindrical or spherical coordinates. The constant ν represents the order of the Bessel functions, which are found as solutions to the Bessel's equation. The constant ν can take any real numbered value although for cylindrical problems the order of the Bessel function is an integer value ($\nu = n$) and spherical problems the order is of half integer value ($\nu = n + 1/2$).

2.1.2 | Bessel Functions

Since Bessel's equation is a second-order differential equation; the general solution can be written in the form of as the addition of two linearly independent solution,

$$y = AJ_\nu(x) + BY_\nu(x) \tag{2}$$

where $J_\nu(x)$ is called the Bessel function of the first kind.In the case of integer or positive values of ν it represents the solutions of Bessel's equation that are finite at the origin ($x = 0$).But for negative or non-integer ν the value diverge as x approaches zero [4].

$$J_\nu(x) = \sum_{k=0}^{\infty} \frac{(-1)^k (x/2)^{\nu+2k}}{k!(\nu + k)!} \tag{3}$$

$Y_\nu(x)$ is called the Bessel function of the second kind which are singular at the origin ($x = 0$). The Bessel function of the second kind, $Y_\nu(x)$ is sometimes referred to as Weber or Neumann function (which can be sometimes be denoted as $N_\nu(x)$). It is related to the Bessel function of the first kind as follows for non-integer ν,

$$Y_\nu(x) = \frac{J_\nu(x)cos(\nu\pi) - J_{-\nu}(x)}{sin(\nu\pi)} \tag{4}$$

In order to get the ν values for integer n, limit as a non-integer ν tends to n is taken.

2.1.3 | Modified Bessel Equation

Bessels equation and its solution is valid for complex arguments of x. For the special case of purely imaginary argument, modified Bessel's equation can be obtained through a simple change of a variable in Bessels equation, from x^2 to $-(ix)^2$ (where $i = \sqrt{-1}$).

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + (-(ix)^2 - \nu^2)y = 0 \tag{5}$$

Which is the same as,

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} - (x^2 + \nu^2)y = 0 \tag{6}$$

Thus the equation is called the modified Bessel equation of order ν.

2.1.4 | Modified Bessel Functions

In the same manner as for the Bessel equation, the solutions for the modified Bessel's equation can be written as follows,

$$y = AJ_\nu(ix) + BY_\nu(ix),\ for\ x > 0 \tag{7}$$

or

$$y = CI_\nu(x) + DK_\nu(x),\ for\ x > 0 \tag{8}$$

Here $I_\nu(x)$ and $K_\nu(x)$ are the modified Bessel functions of the first and second kind of the order ν. $I_\nu(x)$ is defined by

$$I_\nu(x) = (\frac{x}{2})^\nu \sum_{k=0}^{\infty} \frac{(\frac{x^2}{4})^k}{k!\Gamma(\nu + k + 1)} \tag{9}$$

The modified Bessel functions of the second kind, denoted $K_\nu(x)$, form a second solution given by

$$K_\nu(x) = (\frac{\Pi}{2})\frac{I_{-\nu}(x) - I_\nu(x)}{\sin(\nu\pi)} \tag{10}$$

The proof can be found in[5]. Fig.1 shows sample Bessel function plots for the Bessel functions and modified Bessel functions of first and second kind. Y-axis represents the Bessel function values for when $\nu \in [0, 6]$. Modified Bessel functions I_ν and K_ν are exponentially growing and decaying functions as shown in Fig.1 (c) and Fig.1 (d) while ordinary Bessel functions J_ν and Y_ν are oscillating as shown in Fig.1 (a) and Fig.1 (b).

2.2 | Grayscale Image Analysis

Grayscale images are used in most image analysis algorithms due to the simplicity of using a single contrast variant, when compared to the 3 variant RGB color representation of the same image. Hence the algorithm presented in this paper also uses grayscale converted video frames. Analysis of these grayscale video frames first and foremost gives an idea about the correlation between contrast change of an image and it's sharpness. Fig.2 (a) to (d) shows four different frames from a video that was taken of an ornamental object with changing focal length of the camera lens. First sub-figure (a) depicts the sharpest of the four frames, where the object is in focus. Figure (b) shows another frame where the focal length of the camera lens is slightly off from the previous point, so the object is not in sharp focus, but majority of the object detail is still visible. The sub-figure (c) shows a frame where the focal length is even more deviated from sub-figure (b) so the image details are very slightly visible. While (d) shows when the image is almost out of focus where details are hardly visible. Fig.2 (e) to (h) shows the grayscale distribution plots of the voxel [1] intensity values of each frame (a) to (d) respectively. The values on the plots shows the grayscale voxel intensities along the horizontal voxel array shown in each sub-figure (a to d) by a green line on the middle of each frame. The y-axis of each plot shows voxel intensity of the grayscale voxel array that spans from 0 to 255. Where 0 marks the darkest (black voxels) and 255 marks the white voxels that are the brightest. X-axis shows the voxel index that goes from left to right for each frame (1 to 300 voxels, since these are 300*300 voxel images). The peaks of the plots are marked with red inverted triangle while the valleys of the plots are marked with green triangles. When examining the 4 plots, it is evident that number of

[1] voxel is 3D representation of a pixel where x,y and time represents the 3 dimensions

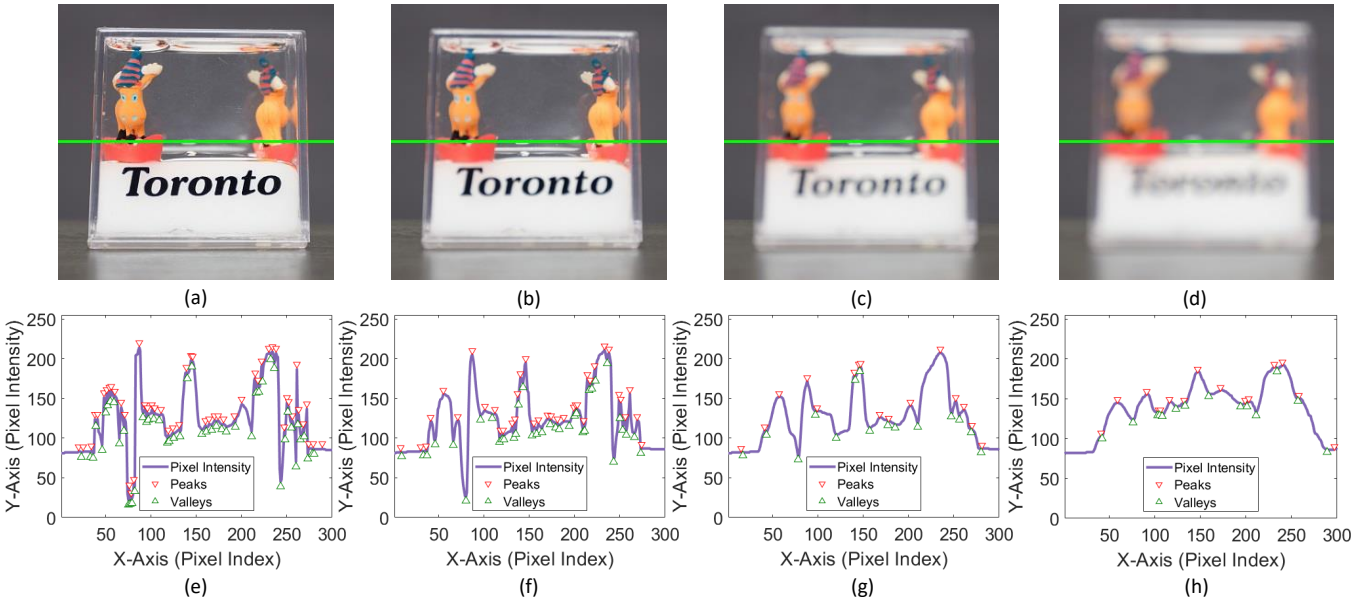
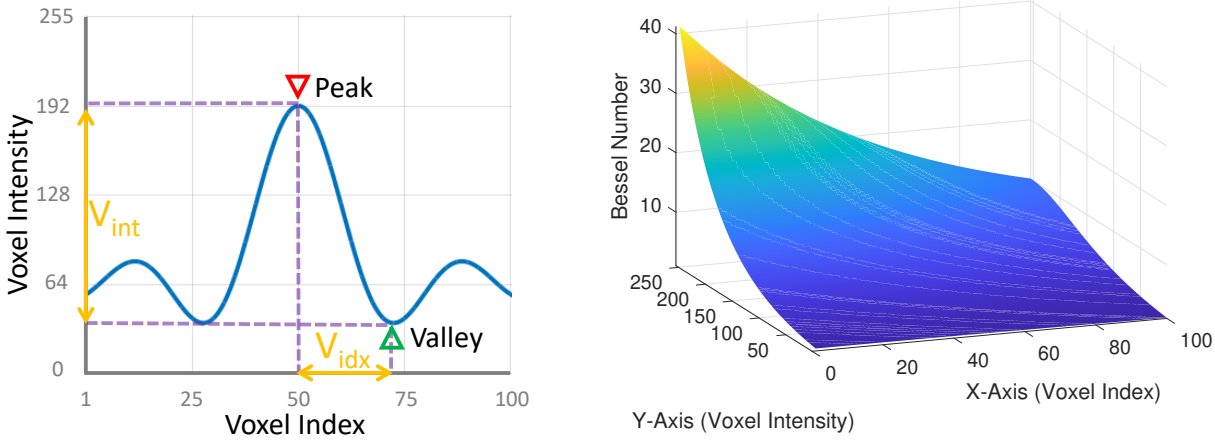


FIGURE 2 Change in the grayscale voxels with image sharpness

peaks and valleys decrease as the object goes out of focus, also the intensity difference between peaks and valleys reduce. The main idea of this paper is to use Bessel functions as a way to measure this contrast change in the video frames, hence measure the change in sharpness of objects in a video frame.

3 | METHOD



(a) Example input array for the algorithm that shows the calculation of voxel intensity and index difference (V_{int} and V_{idx}).

(b) Plot showing Bessel value with respect to voxel Index and voxel Intensity, which was used for the examples in this paper (see Eq. 13).

FIGURE 3 Example voxel intensity array and 3D surface plot of the Bessel matrix.

This section of the paper elaborates the methodology as to how the Bessel functions are used in order to analyze the video frame sharpness.

3.1 | Measuring Intensity Differences

As mentioned in the previous section the main goal of the paper is to measure the changes in the grayscale voxel intensity values using Bessel functions. Fig.3 (a) shows an example where the grayscale intensity values are changing in a voxel array that is 100 voxels in length. Grayscale voxel values can vary from 0- 255. Y-axis represents the voxel intensity values while the X-axis represents the voxel location in the array. The voxel distance between a peak and a valley (or vice versa) is called the $Voxel\ Index\ Difference$,

$$Voxel\ Index\ Difference\ (V_{idx}) = \ |peak\ index - valley\ index| \tag{11}$$

while the absolute value of difference between the peak and the valley of voxel values are called the $Voxel\ Intensity\ Difference$,

$$Voxel\ Intensity\ Difference\ (V_{int}) = \ |Array\{peak\,(i)\} - Array\{vlley\,(i)\}\,| \tag{12}$$

Next step is to use the calculated $Voxel\ Index\ Difference$ and $Voxel\ Intensity\ Difference$ to obtain a single value that can represent the image sharpness. This relationship between input and output is built using Bessel functions. Hence this calculated value is called the Bessel value.

3.2 | Bessel Plot

The relationship between the Bessel value and Voxel Intensity Difference and Voxel Index Difference depends on the type of the Bessel function that is used. For the examples in this paper, the modified Bessel functions of the second kind were used (See equation 10), where $\nu = 0$, to generate the relationship. The exact relationship is given in equation 13 where a scaling factor $m = 3$ was used. The purpose of the scaling factor is to minimize the discrepancies that arise due to noise in the images. With a higher scaling factor the output Bessel value will change drastically when the peaks and valleys increase but with the cost of sensitivity to changes in low peaks and valley areas in the image. Fig.3 (b) show the 3D plot that depicts the relationship for Bessel value as given in equation 13.

$$Bessel\ Value = 1 + \frac{m\sqrt{{V_{int}}^2 + {V_{idx}}^2}}{\sqrt{Array\ Size^2 + 255^2}} \tag{13}$$

3.3 | Example Calculation of Bessel Value

Fig.3 (a) shows an example for the calculation of Voxel Intensity Difference, V_{int}, and Voxel Index Difference, V_{idx}. In the given example the values start from 60 and then increase and decrease along the array, the maximum value is 192. Shown in the example is the calculation of a single V_{int} and V_{idx}. Here $V_{int} = 192 - 40 = 152$ and $V_{idx} = 75 - 50 = 25$. So the Bessel value for this peak and valley calculation can be calculated using equation 13 which is $Bessel\ Value = 1 + 3 * sqrt(152^2 + 25^2)/sqrt(100^2 + 255^2) = 2.6872$. In the example shown in Fig. 3 (a) Bessel values are calculated between each valley and peak, and then between each peak and valley. So in this example total of 6 Bessel values are calculated for the given plot. The Bessel value for the array (or sensor, as explained in section 3.6) is the summation of all the 6 values.

3.4 | Bessel Value Algorithm.

The algorithm to obtain Bessel values are given in pseudo code in algorithm 1. Here the input array ($arrM$) of grayscale voxels has m values (size m) is given to the function and a Bessel Value is obtained as output. The algorithm first search for peaks and valleys of the input array. The obtained valley and peak locations distance as well the corresponding grayscale voxel intensity values are measured using the Bessel matrix (or $bMat$).

Fig.3 (b) shows the 3D surface plot of matrix 'bMat' given as an input variable in the Algorithm 1. Here Z-Axis represents the Bessel value for Voxel Index Difference (X-axis) and Voxel Intensity Difference (Y-axis).

The relationship between Bessel value with voxel index and intensity can be altered by using different variations of Bessel functions. This can be improved to change according to the various objects that needs to be recorded using the camera. For

Algorithm 1 Bessel Value Algorithm

input : Grayscale voxel array arrM
output: Bessel value for the array besselV.

```
function besselV = besselValue (arrM,bMat)
PeakVal ← Find peaks of arrM
VlyVal ← Find valleys of arrM
SmplIdx ← Find indexes of peaks and valleys arrM
besselV = 0
for i ← 1 to length(SmplIdx)-1 do
    vxIndex =abs(SmplIdx (i)-SmplIdx (i+1))
    vxIntensity =abs(arrM (SmplIdx (i))-arrM (SmplIdx (i+1)))
    if isnan(vxIndex)||isnan(vxIntensity)||vxIndex ==0||vxIndex ==0 then
        besselV = besselV
    else
        besselV +=bMat (floor (|vxIntensity |),floor (|vxIndex |)
```

an example macro photographs can use a different variation of the Bessel function when compared to a portrait image. Hence different aspects of the Bessel matrix will allow different Bessel values for different situations.

3.5 | Time Complexity Analysis

Fig. 4 show the time complexity analysis plot for the algorithm given by Alg.1. In order to calculate the theoretical time complexity values, several assumptions were made. The time taken for built-in functions were not considered; for example time taken to import the video frame, save the triangulated frames, initializing variables, inner working of loops etc. Addition, Subtraction, Multiplication, Division were taken as 4 different calculations. Furthermore, allocation of values and array search function were considered to be a single calculation each.

The theoretically obtained time complexity in terms of big O notation was $k_1m + k_2n$ where m is the size of the input array ('arrM' in Alg.1 or the sensor size) and n is the number of peaks and valleys in the array values. The value of k_1 and k_2 was experimentally found to be $2.3e^{-8}s$ and $2.5e^{-6}s$ respectively. In order to obtain the actual time complexity plot, a *sin* array with different peaks and valleys were given as input to the algorithm. To plot the two graphs in the same plot, a base value of $8e^{-4}s$ was added to all the theoretical plots. This represents the time taken to load the variables as well as other neglected functions in the theoretical calculation of time.

In Fig.4 theoretical graph is shown in dotted lines. The actual time complexity is shown using the solid lines. The X-Axis shows the array size or the sensor size while the Y-Axis shows the time taken in seconds. The different plots in different colors corresponds to the different number of peaks and valleys; and are shown in the plot legend. When number of peaks and valleys are increased there is a clear indication of increased computational time. By looking at the plots it is evident that the theoretical and the practical plots follow each other closely, but there are small differences that can be caused due to background functions of the computer affecting the processing of the scripts. For this paper, a fixed sensor size of 100 was used, which reduces the big O notation value of the time complexity to, a linear complexity of n.

3.6 | Implementation

The summery of implementation of the Bessel value algorithm is shown in Fig.5 . When processing the video a grid was used to analyze each video frame, that contains 10 rows and 19 columns. The main reason for using a gird was because of it's flexibility in altering where the image sharpness can be analyzed. Furthermore, changing the properties of the grid (eg. size and location) it is possible to speed up processing or analyses the video more thoroughly. Each video frame in the examples used for this paper is in full HD resolution (1080*1920 voxels). Each block in the grid is called a sensor. A single sensor is 100*100 voxels. So each frame has 190 sensors with voxel values that varies from 0 to 255. Each sensor is analyzed both vertically and horizontally and the average value is taken as the Bessel value for the sensor for that specific frame. In Fig5 these two vertical

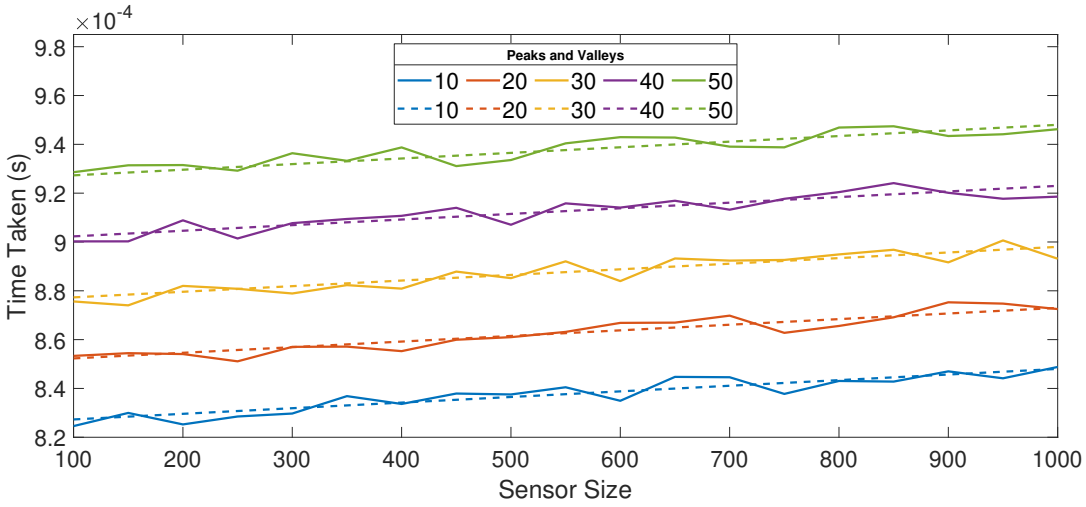


FIGURE 4 Time Complexity Analysis of the Algorithm

and horizontal voxel arrays that analyzed are shown by purple lines on the sensor. These are given as input to the Bessel value algorithm one after the other. In order to increase the accuracy the number of analyzing lines (or arrays) can be increased, for example 3 horizontal lines and 3 vertical lines per sensor. But this is at the expense of the processing time. So this method allows the flexibility of choosing between faster processing time vs higher accuracy. One example for faster processing time is beneficial when taking photograph or video of fast moving subjects like in sports. While accuracy is more important in work like macro photography. Hence the adaptability of the method very high and can be utilized depending on the situation.

The whole video is processed frame by frame using this method and the Bessel value for each sensor for each frame is saved in a matrix for comparison. This matrix is called Bessel Value Matrix (or BVMat) and is of the size ($frameNo * 10 * 19$).

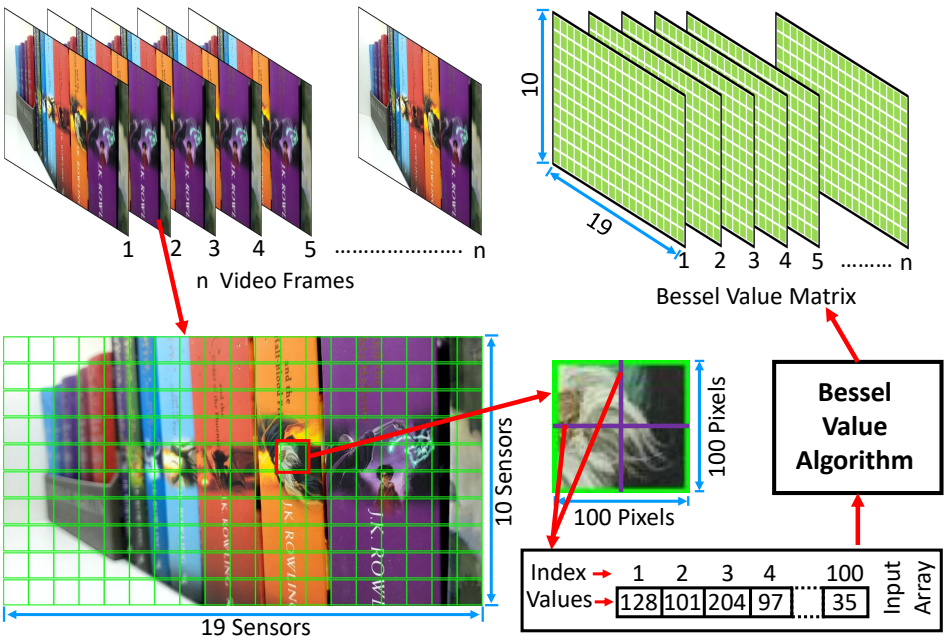


FIGURE 5 The implementation of the Bessel Value algorithm on a video frame.

In the 2^{nd} processing phase for each frame a matrix called Bessel Value Difference (or BV_{dif}) matrix is calculated to find the maximum normalized Bessel value for each video frame. Equation 14 shows the calculation of the value in row i and column

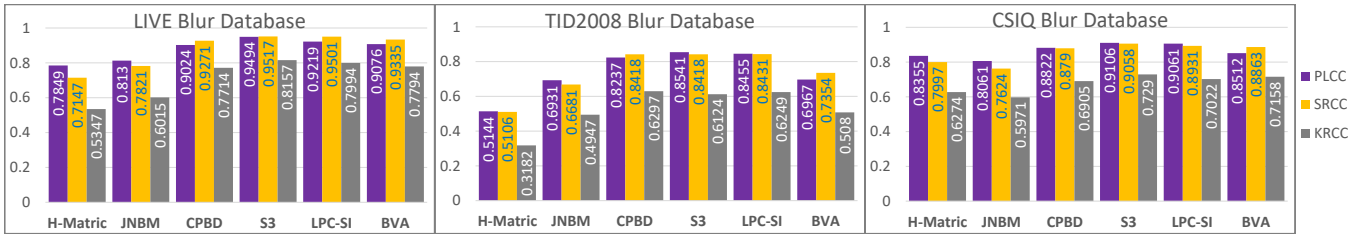


FIGURE 6 Performance evaluation of over three databases.

j of the m^{th} frame in the video. The centroid of the sensor, with the maximum value in this matrix for each frame, is used as the generating point of the barycentric nucleus clusters that are used as a method to display the sharpest area in the frame[6,7]. Grayscale image corners has been used to generate key points, in order to draw the vortexes. Each sensor generate maximum of 4 key points. If the sharpness of the underlying image is low, no key points will be generated for that sensor.

$$BV_{dif(m,i,j)} = \frac{BVMat_{(m,i,j)} - \min\limits_{k\in n}(BvMat_{(k,i,j)})}{\max\limits_{k\in n}(BVMat_{(k,i,j)}) - \min\limits_{k\in n}(BVMat_{(k,i,j)})} \tag{14}$$

3.7 | Performance Evaluation

Analysis of the performance evaluation is important when comparing algorithms to understand where they stand among other algorithms. In this paper performance as well as processing time was compared with 5 other algorithms that are well known and are available online.

3.7.1 | Performance of Image Databases

A standard method of performance analysis for an image sharpness algorithm is to compare the results for no-reference quality blurred image databases. Three publicly available image databases have been analyzed using the algorithm in this paper and the results are compared with the values that were given with the databases. The databases that were used include LIVE2 with 175 images[8], TID2008 with 96 images[9] and CSIQ with 150 images[10].

Each database contains either DMOS (Differential Mean Opinion Score) or MOS(Mean Opinion Score) for the images that they contain. For each set of obtained values PLCC (Pearson's Linear Correlation Coefficient), SRCC (Spearman Rank-order Correlation Coefficient) and KRCC (Kendall's Rank-order Correlation Coefficient) were calculated to find the correlation with the either DMOS or MOS values. The results and are given in the plots in Fig.6 .

When comparing the scores for the BVA (Bessel Value Algorithm), no logistics transform was applied. Hence the values shown in Fig.6 for BVA shows direct correlation values.

Five other algorithms were also implemented in order to compare the results. These were H-metric[11], JNBM[12], CPBD[13], S3[14], LPC-SI[15].

3.7.2 | Processing Time Comparison

Table 1 shows the processing time for each algorithm that was considered for comparison. Here 6 of most common image resolutions were used for the processing time, which included 512*512, 1024*768, 1280*720, 1280*960, 1600*1200 and 1920*1080. All the algorithms were implemented in Matlab 2020a and processed using an AMD 2700X processor based desktop PC with 32GB of RAM. BVA shows the time taken for the processing of the whole image as a single input (whole image as a single sensor), while BVA_{100} shows the processing time using a sensor size of 100*100 voxels. For a fare comparison all the scripts were run using single thread Matlab script. Even though the Bessel Value algorithm was implemented to work as multiple threads using parallel processing tool box which gives roughly 10X faster processing time when performed with the PC mentioned earlier. But since parallel processing depends heavily on the implementation and the hardware (The processor core count and RAM) for time comparison single threaded version of the algorithm was used. Each algorithm was run for 100 iterations per image and the time was averaged in order to avoid the effects that usually arise by the background processes of the PC.

Resolution	512*512	1024*768	1280*720	1280*960	1600*1200	1920*1080
H-Metric	0.9650	2.8950	3.3926	4.5234	7.0679	7.6333
JNBM	0.6065	2.4230	2.8395	3.1197	7.0368	7.5997
CPBD	0.7073	3.2075	3.7588	4.3731	10.8920	11.7633
S3	9.5371	28.1034	33.3901	43.5819	67.8597	73.7360
LPC-SI	0.2974	1.2422	1.4557	1.4751	3.1773	3.4315
BVA	0.0018	0.0018	0.0018	0.0019	0.0020	0.0020
BVA100	0.0316	0.0878	0.1051	0.1349	0.2360	0.2385

TABLE 1 Time taken for processing (in seconds)

All the other algorithms were used, in the form they were made available online by their authors. Furthermore, the generation of plots and display outputs of any sort were disabled in all algorithms for fair comparison among each other, since they all use different output and display methods.

3.7.3 | Performance Results Comparison

When looking at the results it is clearly visible that BV algorithm is much faster than the other algorithms that were compared. When comparing the results for the correlation and the processing time, it is evident that even though the Bessel Value algorithm isn't as accurate as some of the algorithms that were compered, it is much faster than all of the compared algorithms. Which makes it a ideal candidate for real time processing with a comparatively high accuracy. As mentioned earlier, the sensors can be modified to use more than one horizontal and vertical arrays for processing, which will increase the accuracy of the algorithm.

4 | RESULTS

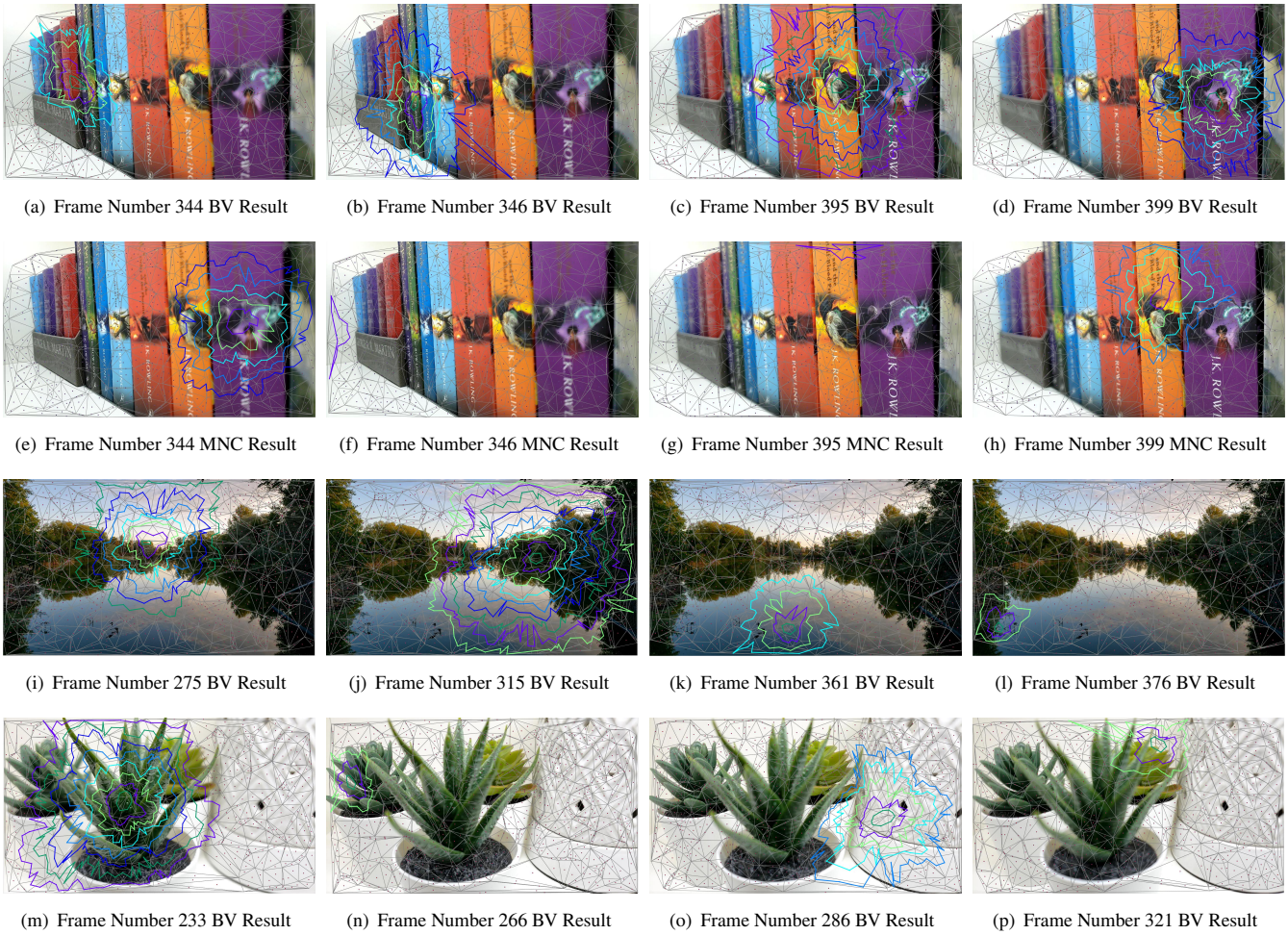
(a) Book Shelf (b) Lake View (c) Pot Plants

FIGURE 7 Sample frames from the three videos that were used.

Three sample videos were analyzed in this paper. The first video is of a book shelf where the camera is pointed towards set of books. The second is of a lake while the third video is of some pot plants. The videos were taken with changing focal length in order to analyze the theory. All three are 30 fps (frames per second) videos of 1920*1080 resolution. Fig.7 shows three sample frames from each video.

4.1 | Example 1: Book Shelf Video

Given in Fig.8 is four sample video frames from the book shelf video. They represent frames 344, 346, 395, 399 in the video (Average length of a video is 30 seconds or 900 frames, frame number is given as an indication to the time gap between the frames). Figures 8 (a), 8 (b), 8 (c) and 8 (d) represent the output generated using Bessel value algorithm while sub figures 8 (e), 8 (f), 8 (g) and 8 (h) represents the results for MNC (Maximum Nucleus Cluster) based vortexes [16].



(a) Frame Number 344 BV Result (b) Frame Number 346 BV Result (c) Frame Number 395 BV Result (d) Frame Number 399 BV Result

(e) Frame Number 344 MNC Result (f) Frame Number 346 MNC Result (g) Frame Number 395 MNC Result (h) Frame Number 399 MNC Result

(i) Frame Number 275 BV Result (j) Frame Number 315 BV Result (k) Frame Number 361 BV Result (l) Frame Number 376 BV Result

(m) Frame Number 233 BV Result (n) Frame Number 266 BV Result (o) Frame Number 286 BV Result (p) Frame Number 321 BV Result

FIGURE 8 Resulting video frames from three sample videos.

Examining the 4 video frames that use BVA, it is evident that the nucleus of the vortex falls on the sharpest area of the frame. Which yields much better results when compared to the MNC based vortex clusters [16]. Even though, MNC based vortexes gives somewhat intuitive results in some cases, when analyzing sharpness on an image BVA based vortexes are much suited and more consistent.

4.2 | Example 2: Lake View Video

Next resulting frames are from lake view video. They are given by Fig.8 (i),8 (j),8 (k) and 8 (l) which are 275, 315, 361 and 376 frames of the video respectively. When examining frames the focal point seems to change from far to near. When examine the video frames it is evident that this correlation with the obtained results. In Fig.8 (i) the vortexes are based on a location that is far in the frame. When the frame number increases, following the image focus, the generated vortex comes closer to the foreground where the lake banks are. Furthermore, using vortexes to highlight the sharpest area in a video frame is more effective than using rectangles or squares. This is mainly due to the fact that generation of vortex points depends on the underlying image it self. So vortexes tend to contour according to the image shapes themselves. One good example is the upper portion of the vortexes in Fig.8 (i), it can clearly be seen that they contour in the shape of sky in the horizon.

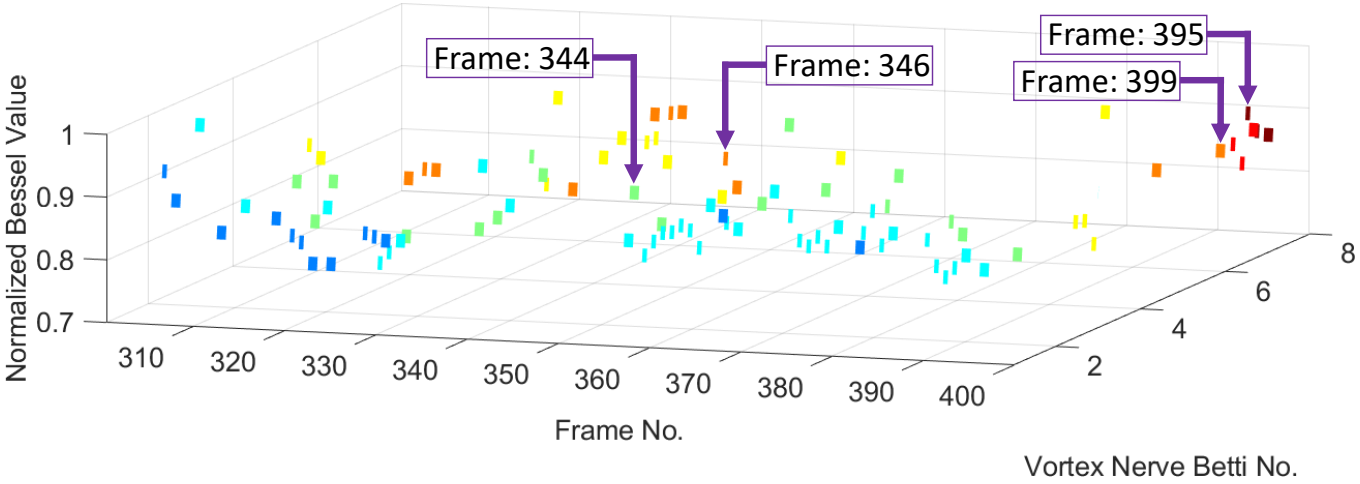


FIGURE 9 Normalized Bessel Number Vs Betti Number Vs Frame, 3D Barcode.

4.3 | Example 3: Pot Plants Video

There results from the pot plants video is given by Fig.8 (m), 8 (n), 8 (o) and 8 (p). In the frame 233 the focus is on the middle of the frame where the main pot plant is. The generated vortex covers the majority of the frame. It is evident that the areas that are not covered by the vortexes are soft and blurry. Likewise the vortexes follow the areas of the frame where that are in sharp focus.

4.4 | 3D Barcodes and Betti Numbers

Fig.9 shows the 3D barcode for Normalized Bessel Number Vs Betti Number Vs Frame. Here Betti number β_1 was used which is the count of cycles in the vortex [17]. 3D barcodes were introduced as an efficient and simple way of representing video frame metrics [6]. The barcode represent the video frames 300 to 400 from the bookshelf video. In each video frame the Betti number β_1 is plotted using a short line. In Frame 344 shown by Fig.8 (e) the Betti number is equal to 4, since there are 4 vortexes. Same way frames 346, 395 and 399 have Betti numbers 6, 8 and 6. A clear observation is that, low Betti number means the sharpest areas of the frame is closer to the edges of the video frame. In the same manner, high Betti number means the sharpest areas are closer to the middle of the frame. Hence Betti number is a good indicator of the location of the sharpest area of the image. This is very helpful in a scenario where video re-framing is concerned. Betti number can be used to find the sharpest areas of the video hence re-frame the video to represent higher detailed zoomed frames.

5 | CONCLUSION

This paper introduces an application of Bessel functions, that can be used to find the sharpest area of a video frame. Major contribution of this work includes measuring the changes in video frame contrast using modified Bessel functions of the second kind as a way to measure the sharpness of the frame. The introduced algorithm uses a grid based frame analysis that is highly flexible in application while barycentric vortexes has been used to display the sharpest areas in video frames. Results from three sample videos has been provided along side performance comparison with well known other image sharpness measuring algorithms to demonstrate the effectiveness of the proposed algorithm. Furthermore, time complexity and performance analysis has been conducted and the results has been presented. The paper also ties in the obtained results with previous work by generating a 3D barcodes to present the obtained results. In a future follow up of this paper different types of Bessel functions are to be analyzed as to where they can be used optimally for recording of different subject matter.

References

1. Bossuyt S.. Optimized Patterns for Digital Image Correlation. In: The Society for Experimental Mechanics, Inc. 2013.
2. Vasil'ev K.K.. Digital Processing of Sequences of Multizonal Images. *Pattern Recognition and Image Analysis.* 2008;18(3):376–380.
3. Barnett A.H.. How exponentially ill-conditioned are contiguous submatrices of the Fourier matrix. *arXiv.* 2020;2004(09643v3):1–24.
4. Bowman Frank. *Introduction to Bessel Functions, 1958.*
5. Abramowitz Milton, Stegun Irene A. Handbook of mathematical functions with formulas, graphs, and mathematical table. In: National Bureau of Standards Applied Mathematics series 55 1965.
6. Don Arjuna PH, Peters James F, Ramanna Sheela, Tozzi Arturo. Topological View of Flows Inside the BOLD Spontaneous Activity of the Human Brain. *Frontiers in Computational Neuroscience.* 2020;14:34.
7. Peters James F. Computational Geometry, Topology and Physics of Visual Scenes. In: Springer 2020 (pp. 1–85).
8. Sheikh HR. LIVE image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality.* 2005;.
9. Ponomarenko Nikolay, Lukin Vladimir, Zelensky Alexander, Egiazarian Karen, Carli Marco, Battisti Federica. TID2008-a database for evaluation of full-reference visual quality assessment metrics. *Advances of Modern Radioelectronics.* 2009;10(4):30–45.
10. Larson Eric C, Chandler DM. *Categorical image quality (CSIQ) database.* 2010.
11. Mansouri Azadeh, Aznaveh Ahmad Mahmoudi, Torkamani-Azar Farah, Jahanshahi J Afshar. Image quality assessment using the singular value decomposition theorem. *Optical Review.* 2009;16(2):49–53.
12. Ferzli Rony, Karam Lina J. A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB). *IEEE transactions on image processing.* 2009;18(4):717–728.
13. Narvekar Niranjan D, Karam Lina J. A no-reference image blur metric based on the cumulative probability of blur detection (CPBD). *IEEE Transactions on Image Processing.* 2011;20(9):2678–2683.
14. Vu Cuong T, Phan Thien D, Chandler Damon M. S3: A Spectral and Spatial Measure of Local Perceived Sharpness in Natural Images. *IEEE transactions on image processing.* 2011;21(3):934–945.
15. Hassen Rania, Wang Zhou, Salama Magdy MA. Image sharpness assessment based on local phase coherence. *IEEE Transactions on Image Processing.* 2013;22(7):2798–2810.
16. Don A.P.H., Peters J.F.. Ghrist Barcoded Video Frames. Application in Detecting Persistent Visual Scene Surface Shapes captured in Videos. *Theory and Applications of Mathematics & Computer Science.* 2019;9(1):14–27. https://uav.ro/applications/se/journal/index.php/TAMCS/article/view/182/150.
17. Zomorodian A.J.. Computing and Comprehending Topology Persistence and Hierarchical Morse Complexes. PhD thesisUniversity of Illinois at Urbana-ChampaignGraduate College2001. supervisor: H. Edelsbrunner, 199pp.