

# *Rehabilitation Low Vision Algorithm*

*For People with central or multiple losses of vision*

Habib M. Fardoun, Lorenzo C. González

Information Systems Department  
King Abdulaziz University (KAU)  
Jeddah, Saudi Arabia  
{hfardoun, lgonzalez}@kau.edu.sa

Abulfattah S. Mashat

Information Technology Department  
King Abdulaziz University (KAU)  
Jeddah, Saudi Arabia  
msmashat@kau.edu.sa

**Abstract**— This article focuses on the rehabilitation of people with low vision, concretely on people with central or multiple loss of vision, and the use of new technological devices to assist them in their daily life by improving their ability to see in a better way the world around them, either virtual or real. This particular case refers to users with impaired vision, and the use of digital image processing algorithms to show the hidden areas that the eye can't see, by moving that information to a visible area of the user. For it, we describe the Low Vision algorithm, which is based on “Bresenham algorithm” as a solution to help rehabilitate people with those types of vision problem.

**Keywords**(*Rehabilitation, digital image processing, Bresenham algorithm, human computer interaction, Low Vision algorithm, eye tracking*).

## I. INTRODUCTION

Vision problems occur throughout life and old age causes many of them. In others, the causes are genetic or due to accidents. People who suffer with vision problems have difficulty adapting to everyday situations [1]. Often, the problems related to sight are irreversible and it is necessary to rehabilitate the person's sight, reducing the change in vision as much as possible in order for them to continue living life in the best way possible [2] [3] [4]. An individual has low vision when he has a diminution of the visual functions, not recoverable by treatment and/or refractive correction. Any person whose visual acuity is between 0.005 and 0.3, in the better eye and/or visual field of less than 20° in the better eye has a problem related with low vision, in this paper we are going to focus on central and multiple losses. These people keep some useful vision, known as residual vision, which can be improved or enhanced by the use of optical and non-optical aids, and subsequent visual rehabilitation [1].

This article discusses a solution related to low vision problems and central and multiple loss of vision resulting from diseases or injuries, such as macular degeneration, optic atrophy, diabetes, glaucoma or detachment of the retina [5] [6]. Such conditions cause deterioration in eyesight and make regular activities such as looking at a picture or reading a newspaper, more difficult, and sometimes too difficult to continue [7].

To improve this type of vision loss, and reveal the information hidden behind the non-visible areas of the eye, an algorithm has been created for digital image processing, based

on the Bresenham Algorithm [8][9], which shows that a lesser distortion of reality is possible. The new algorithm (Low Vision algorithm) uses the information of non visible areas and merges it with the information of the closest visible area to the first one. Thus, user can see the data hidden by the non visible zone of his eye. This together with an eye motion-tracking algorithm and new technologies, can make life easier for these people, and enable them to continue with daily activities. In this article we focus on using the algorithm on screen devices and by mean of them users with this type of problem can see another reality.

In the following sections we will discuss existing ways to rehabilitate these types of vision problems, the Bresenham algorithm, the Low Vision algorithm, how to embed them in mobile devices and the work that is being done to improve techniques.

## II. STATE OF ART

In relation to the rehabilitation of visually impaired, it is more difficult to find solutions that go beyond items such as spectacles, lenses, or other types of concrete object. There are two types of solutions for people with low vision related to loss of central or multiple visions: optical and non-optical:

- Optical. Uses lenses based on the magnification or minification of the image. Between them we can find:
  - Kepler and Galileo Telescopes. Used for distance vision.
  - Hyperocular glasses, magnifiers, magniplus, etc. Used for near vision.
- Non-optical. Improves visualisation of residual vision without optical aids. We highlight:
  - Filters. Used for problems related to light incidence on the eye and the light spectrum.
  - Tiposcopio. Used for reading and writing to prevent changing line and text dazzle.
  - Large print. Large text size, for easy reading.

There are different types of technique, depending on the category of vision loss. For central field loss, image magnification is used; while for peripheral loss prisms are used, which move lateral images to the centre. Minificators

increase the visual field at the expense of losing some detail from the captured image.

As we are going to apply the algorithm to screen devices, we need software to detect the eye movement because the user doesn't look always at the same place but he moves the eye to areas where he wants to get some information. Various other papers propose software solutions related to the detection of eye movement as a whole or in the pupil in particular. We are going to mention them but the choice isn't final yet. Below we give several examples of solutions for eye movement:

- TrackEye: Real-time tracking of human eyes using a webcam [10]. Software that uses OpenCV libraries to process images that detects both head and eye movement.
- Robust real-time pupil tracking in highly off-axis images [11]. An algorithm, which precisely locates pupil movement using different techniques, some related to the adjustment of ellipses and k-means segmentation.
- Pupil Positioning Algorithm using Circular Hough Transform and Grey Projection [12]. An algorithm that detects pupil position based on edge detection algorithms and adjustment algorithms of ellipses.

### III. PROBLEMS AND SOLUTIONS

Vision is reduced by disease or trauma and people have to adapt their behaviour to overcome this. For example, a person who could see all the details in a picture prior to deterioration in their vision has to look at it several times from different angles in order to try to see it properly.

As it is not possible to see the same amount of detail without some loss of vision over a smaller area of the eye, the use of an algorithm is proposed. This algorithm uses a part of the user's visible area closest to the periphery of non visible zone and merges the pixels of that zone with the pixels on the visible area. Thus, previously hidden detail can be seen. The surface of the eye to be treated suffers a deformation, which due to the loss of information, slightly distorts the reality.

The proposed solution consists of an eye tracking movement algorithm, the Low Vision algorithm, and software, which runs on the user's device. Below are some brief comments about each of these elements:

#### A. Location algorithm eye movement.

This algorithm is used for the detection of eye movement to find out exactly where the user is looking. If there is a change in the direction of the eye, to get a full visual experience, it is necessary to recalculate the Low Vision algorithm. As, after a shift in movement, some areas of the eye clogs up other areas, so that the detail seen is different. There are several approaches available online and various documents that discuss how to detect this type of motion [11] [12] [13]. This algorithm is not the focus of this paper so we do not show its implementation and in addition, we are testing several of them to get the best results.

#### B. Low Vision algorithm.

This algorithm is the focus of this paper and we will explain it in detail. Its function is to improve the eyesight of the user by revealing the previously unseen information. We achieve this by using the Bresenham algorithm to create digital lines between two points.

#### C. Software for Low Vision algorithm.

This software runs continuously on any application on the user's device. It allows the user to see the most detail. It requires calibration to determine areas where the user cannot see. To determine those areas, user will look at a point on the screen and then he will paint (with the finger or mouse) the periphery of said zones. An additional calibration algorithm makes use of eye movement tracking, and determines the limits and eye positions of the user. With it, the person using the device will see the maximum amount of information.

### IV. BRESENHAM ALGORITHM

The Bresenham algorithm was created for drawing lines in raster graphics display devices, such as computer monitors, tablets and projectors, etc. It determines what pixels are filled in which it depends on the inclination of the angle of the straight line drawn between two previously specified points. Once you have the start and end of the line, we apply the algorithm to draw the remainder of the pixels finishing in a straight line.

This determines the pixels selected to be part of the line between the points  $P_0(x_0, y_0)$  and  $P_1(x_1, y_1)$ . Where we have the points  $P_0(0,0)$  and  $P_1(4,5)$  that correspond to the start and ends respectively, Figure 1 shows the resulting line for this example.

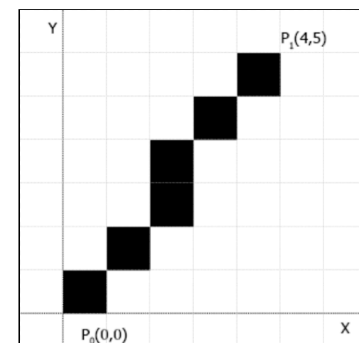


Figure 1. Line drawn using the Bresenham algorithm between the points  $P_0(0,0)$  and  $P_1(4,5)$ .

This algorithm forms the basis of the creation of the Low Vision algorithm proposed in this paper. Its function is to determine each of the pixels constituting the lines that arise after joining the centre with each of the points of a perimeter formed by a non-visible area.

### V. LOW VISION ALGORITHM

The Low Vision algorithm uses the pixels that the user cannot see and displays the information in the visible zone by merging with the pixels in that area. This process is carried out

using the Bresenham algorithm and produces a visual distortion and loss of information because of showing a greater amount of visual data to the user. It allows previously hidden details to appear in the useful zone of the eyepiece, which increases the perception of the surrounding elements.

To achieve this, a specific procedure is required that prepares the workspace for further calculations that shape the algorithm. The process is as follows:

*A. Division into quadrants.*

The area that the user cannot see (Figure 2.), is located in the centre of said area, and the surface is divided into four by the X and Y-axis whose intersection is the said centre.

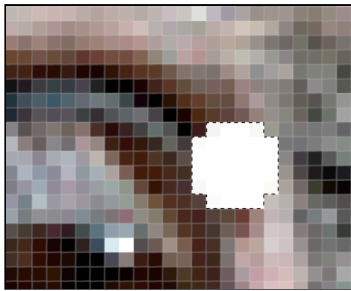


Figure 2. Example of a hidden area to the user's eye

It should be noted that in each quadrant there should be the same number of pixels (there may be a slight variation if the area of the hidden zone is not divisible by four, see Figure 3.).

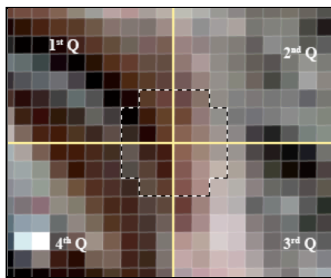


Figure 3. Division into quadrants of the non-visible zone.

Once done, the calculations are performed quadrant by quadrant.

*B. Location of the target surface.*

The work area is calculated for each quadrant, which house the pixels resulting from the merger of the area visible pixels and the non-visible area pixels. The areas corresponding to the peripheral pixels of the hidden area are determined and result in the doubling of the value of each component of the peripheral point associated to the pixel. That is, if the pixel is the (-2, 3) then we have the area associated with the pixel (-4, 6) which corresponds to that contained between said point and the axes. After calculating all areas, binding takes place, whose result minus the hidden area corresponds to the working area (Figure 4.).

Although the next image shows the pixels corresponding to the hidden area, we will not display these pixels to the user. We only show them here to aid understanding of future calculations of this algorithm.

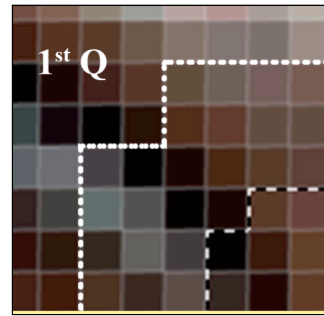


Figure 4. Workspace of the first quadrant.

*C. Determine the resulting lines using Bresenham's algorithm.*

Once we have the workspace, we calculate all lines that constitute it. We make this calculation from the central pixel (pixel nearest to the intersection of the axes) to each of the pixels, which form the perimeter of the area. Peripheral pixels are formed by the pixels, which do not have all their edges along interior pixels or axes.

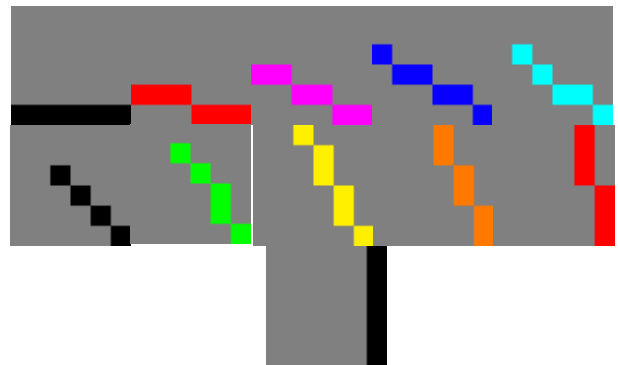


Figure 5. Resulting Bresenham lines.

Thus, each pixel belongs to a certain line. Although, there are cases in which a pixel may be part of two different lines. For this reason, each object "pixel" of the lines should save information related to what line it belongs to, and its position within it, starting from the perimeter, see Figure 5.

*D. Calculation of the pixels of the workspace. Low Vision algorithm.*

For each line of the above, individual calculations for the perimeter pixels to the end of the workspace (end of visible area) are started. For this, each pixel in a position  $p$  within the line, is determined by the average of the values of the pixels whose position is  $(p \times 2)$  and  $(p \times 2) - 1$  within the same line. In case a pixel  $P$  belongs to more than one line,  $P \in L_1$  and  $P \in L_2$ , whose positions would be  $p_1 \in L_1$  and  $p_2 \in L_2$ , the value of  $P$  will be the mean among the pixels whose positions are  $(p_1 \times 2)$ ,  $(p_1 \times 2) - 1$ ,  $(p_2 \times 2)$  and  $(p_2 \times 2) - 1$ . We apply the above

calculation to the pixel values, described in Figure 6., the results of which are shown in Figure 7. For cases where there are more pixels in the working area than in the hidden area, the pixels of that zone, which are left without pixels from the hidden area, will have the nearest pixel to the intersection of the X and Y axes as a reference.

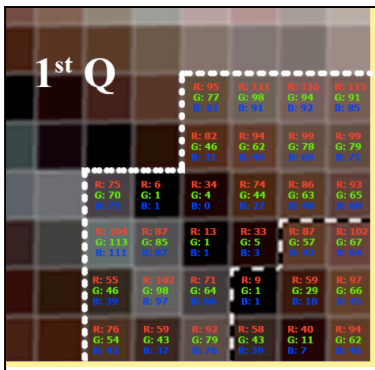


Figure 6. Initial Pixel values.

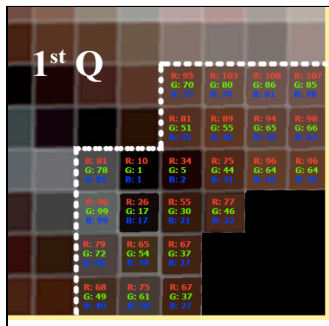


Figure 7. Pixel values after the algorithm.

Using this process, the information from the hidden area will pass to the visible area. Figure 8., shows the difference between how someone without the described eye problems sees the image, and how the same person sees it after applying the Low Vision algorithm.

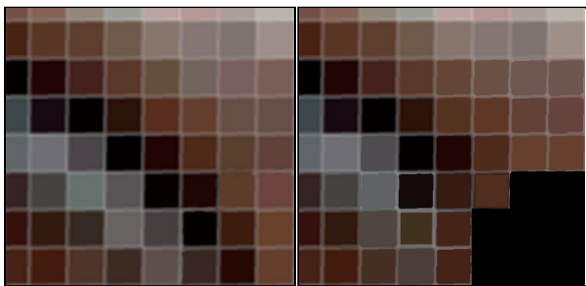


Figure 8. From left to right, image seen by a user without visual problems and image seen by a user with low vision problems and with the algorithm applied.

A breakdown of the algorithm is described below:

- *GetBLines (ListPixel, Pixel)*. Returns a list *BLines*, which has the corresponding peripheral pixel in its first position and the central pixel in last position. Each pixel will be an object *BPixel* that contains information

about the colour. If it has been used for the calculation and is within the work area, its position in relation to the axes and a list of references to pixels should be taken as a basis to recalculate the original.

- *BPixel.getRefPixels()*. Returns a list of *BPixel* to be used to calculate the new colour of the original pixel.

Once the algorithm has finished running and the resultant image produced, we repeat the same procedure for the other three quadrants. By the end of the process, the resultant image displays the hidden information that the visual field of the user cannot see. To help the reader to recognize the effects of the developed algorithm more graphically, we describe another example.

Using the image in Figure 9. as a base, and the perspective that a user would have with the hidden area shown in Figure 10., the Low Vision algorithm is applied.

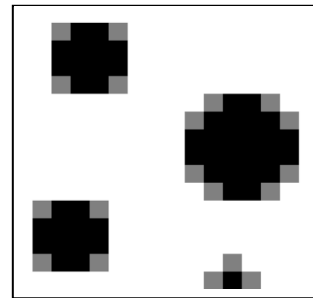


Figure 9. Original image.

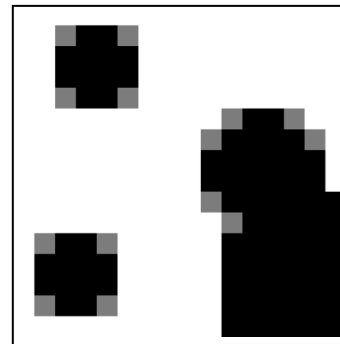


Figure 10. Image of a user with eye problems related to low vision.

We display here in Figure 11., the result obtained after the application of the algorithm and can see how it provides previously hidden information to the user.

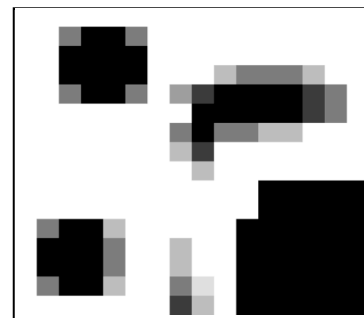


Figure 11. Resulting image after applying the Low Vision algorithm.

We can say that from the user's perspective, this solution provides information about the surrounding world that was previously hidden.

## VI. DEPLOYMENT

When we apply the Eye Movement Detection and Low Vision algorithms together with the associated software, they should help the user to improve his experience with the device. Once the full set of the Ocular Movement Detection algorithm and Low Vision algorithm software is available, it must be installed and be run continuously on the users' machine to get the complete experience. After installation, an initial calibration should be performed that consists of looking at a fixed point on the screen from a certain distance, and then the hidden zones are painted in order for the program to recognize such areas. Once the non-visible zones are determined, it should be calibrated to the eye detection system, looking at the same point in the previous phase, and by moving the eyes to different parts of the device. After completing the installation procedure, it will be possible to perform any activity with the device, including viewing and taking photos, web browsing etc.

## VII. CONCLUSIONS AND FUTURE WORK

People with low vision problems caused by age, disease, or trauma, find their daily lives more complicated by their poor vision. In this paper, we propose a solution to this problem that can help to rehabilitate those people so they can continue with the activities they did prior to the deterioration in their vision. Simple activities such as reading a newspaper or using the internet can be improved with this solution. To achieve this, we have developed the Low Vision algorithm, which uses an area of the user's visual zone and shows details that were previously hidden to the user. Like any research project, it is not without problems that we can strive to improve. Therefore, we are working on improving the functionality and user experience by paying more attention to the following:

- Algorithm for readings. We are working on an algorithm that uses the Low Vision algorithm, and substitutes Bresenham's lines with horizontal lines. This algorithm of horizontal lines improves the visualisation of texts by omitting vertical deformations.
- Improvement on eye movement detection. Further work on the synchronization between eye and head movement, and the Low Vision algorithm. If either of

them suffers a mismatch, the user experience plummets, causing discomfort.

The proposed algorithm in this paper is being validated with 50 people with low vision problems. Where we have divided them in two groups of 25 people each. The first group apply the traditional rehabilitation used in the King Abdulaziz Hospital; and the second experimental group applies the proposed algorithm. In this moment we are in the results collecting phase, which are showing promising results regarding the algorithm use.

## REFERENCES

- [1] J. Ballemans, G.A. Zijlstra, G.H. van Rens, J.S. Schouten and G.I. Kempen "Usefulness and acceptability of a standardised orientation and mobility training for partially-sighted older adults using an identification cane," *BMC Health Serv Res*, 2012, pp. 12:141.
- [2] Zijlstra GA, Ballemans J, Kempen GI. Orientation and mobility training for adults with low vision: a new standardized approach. *Clin Rehabil*. 2013 Jan;27(1):3-18.
- [3] Virgili G, Rubin G. Orientation and mobility training for adults with low vision. *Cochrane Database Syst Rev*. 2010 May 12;(5)
- [4] Lamoureux EL, Pallant JF, Pesudovs K, Rees G, Hassell JB, Keeffe JE. The effectiveness of low-vision rehabilitation on participation in daily living and quality of life. *Invest Ophthalmol Vis Sci*. 2007 April;48(4):1476-82.
- [5] A. Chonsky, "Current Trends in Occupational Therapy Low Vision Rehabilitation", School of Occupational Therapy Theses, 2012
- [6] Singer MA, Amir N, Herro A, Porbandarwalla SS, Pollard J. Improving quality of life in patients with end-stage age-related macular degeneration: focus on miniature ocular implants. *Clin Ophthalmol*. 2012;6:33-9.
- [7] Fahad Al-Wadani, Rajiv Khandekar, Muneera A. Al-Hussain, Ahmed A. Alkhawaja, Mohammed Sarfaraz Khan, and Ramzy A Alsulaiman. Magnitude and Causes of Low Vision Disability (Moderate and Severe Visual Impairment) among Students of Al-Noor Institute for the Blind in Al-Hassa, Saudi Arabia. *Sultan Qaboos Univ Med J*. 2012 February; 12(1): 62-68.
- [8] González, E. Usón, Sobrado Calvo, P., Avellaneda Guirao, M.I., López, M. . Baja visión y rehabilitación visual: una alternativa clínica. Nº:38. *Clinica Universitaria de Visión Integral (CUVI)*. Universidad de Murcia.
- [9] Bresenham, J. E., (1 January 1965). "Algorithm for computer control of a digital plotter". *IBM Systems Journal* 4 (1): 25-30. doi:10.1147/sj.41.0025.
- [10] Watt, Alan (2000). «Rasterizing edges». *3D Computer Graphics* (3ª edición). p. 184. ISBN 0-201-39855-9.
- [11] TrackEye : Real-Time Tracking Of Human Eyes Using a Webcam. <http://www.codeproject.com/Articles/26897/TrackEye-Real-Time-Tracking-Of-Human-Eyes-Using-a> (Accessed: January 2013)
- [12] Swirski, Bulling, Dodgson. 2012. Robust real-time pupil tracking in highly off-axis images.
- [13] Milad Soltany, Saeid Toosi Zadeh and Hamid-Reza Pourreza. 2011. Fast and Accurate Pupil Positioning Algorithm using Circular Hough Transform and Grey Projection.