

# LVRA: Low Vision Reading Algorithm

Mohamed A. Mashat  
King Abdulaziz University  
Jeddah, Saudi Arabia  
mamashat@kau.edu.sa

Habib M. Fardoun  
King Abdulaziz University  
Jeddah, Saudi Arabia  
hfardoun@kau.edu.sa

Lorenzo Carretero González  
King Abdulaziz University  
Jeddah, Saudi Arabia  
lgonzalez@kau.edu.sa

## ABSTRACT

This paper presents the Low Vision Reading Algorithm for people with low vision problems; Low Vision Algorithm is used to reveal the hidden information under the zones in which the user cannot see correctly. This algorithm improves the obtained result when the device is used to read some type of document or web page. Thus, if it is connected with a ocular movement detection application, it aids people to use the devices in a more comfortable way.

## Categories and Subject Descriptors

D.1.7 [Programming Techniques]: Visual Programming.

F.2.1 [Analysis of Algorithms and Problem Complexity]: Numerical Algorithms and Problems – *Computation of transforms.*

H.5.2 [Information Interfaces and Presentation]: User Interfaces – *Graphical user interfaces (GUI), Interaction styles, Screen design, User-centered design.*

I.1.2 [Computing Methodologies]: Algorithms – *Analysis of algorithms.*

I.3.3 [Computer Graphics]: Picture/Image Generation – *Display algorithms, Line and curve generation, Viewing algorithms.*

I.4.5 [Image Processing and Computer Vision]: Reconstruction – *Series expansion methods, Summation methods, Transform methods.*

J.3 [Computer Applications]: Life and Medical Sciences – *Health, Medical information systems.*

## General Terms

Algorithms

## Keywords

Low Vision Algorithm, Low Vision Reading Algorithm, Bresenham Algorithm, Human Computer Interaction, Rehabilitation, Digital Image Processing.

## 1. INTRODUCTION

In previous research studies [1] we presented results on the ways an algorithm like the LVA (Low Vision Algorithm), can aid

designers in creating systems to support people with low vision problems related to central or multiple losses to rehabilitate. Such support will help them to be able to use mobile devices again losing as less information as possible. [2][3][4]. These problems can be due to diseases or injuries, as for example, macular degeneration, optic atrophy, diabetes, glaucoma or retinal detachment [5][6][7]. Thus, when using this algorithm the hidden data are moved from the non-visible area to the visible area, so that the user loses as fewer amounts of data as possible. However, this transformation provokes a deformation at the user's visibility area. This deformation is not that annoying when is treated with images or other elements. However, there are characters to be treated and have to be read, the situation changes. In this case, the characters would slightly vary in size and position, making reading a more difficult activity, and consequently, requiring the reader to make more effort to perform this task.

For these reasons, a new algorithm is introduced: LVRA (Low Vision Reading Algorithm) is based on the previous LVA, hence, it is modified to support reading a specific document and thus, making this task much easier.

## 2. STATE OF ART: LVA

As presented previously, and due to the pixels utilisation of the non-visible area for joining them with the pixels of the visible area, LVA suggests a deformation, which when reading, can be rather annoying. The following figure (Figure 1) presents such deformation. [9][10]

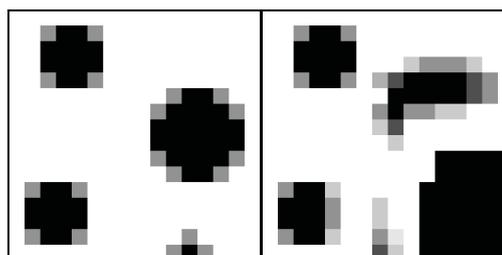


Figure 1. Before and after of applying the LVA algorithm

For the case of using documents, with contents that consist of set of legible characters rather than images or other elements, the obtained result is complicated so to perform the reading task, as presented in Figure 2.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

REHAB 2014, May 20-23, Oldenburg, Germany

Copyright © 2014 ICST 978-1-63190-011-2

DOI 10.4108/icst.pervasivehealth.2014.255254

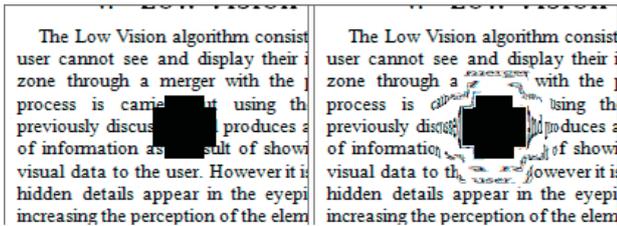


Figure 2. Before and after applying the LVA to a text file

As in the previous figure, although the hidden characters behind the non-visible area are now visible, an effect is produced that makes clear reading rather complicated. For this reason, LVRA is developed to be used for these particular cases where the reading of a specified document is of high priority.

### 3. LVRA: LOW VISION READING ALGORITHM

The Low Vision Reading Algorithm (LVRA) improves the texts' visualization for the reader/user who does not have so many difficulties and the time to elucidate the characters near to the non-visible area. Therefore, the system is used to read horizontal lines, avoiding by this the deformation on the axes and making the texts more understandable. In the Figure 3 below, a non-visible area does not allow the reader to see a whole word, losing part of the meaning that the sentence could have.

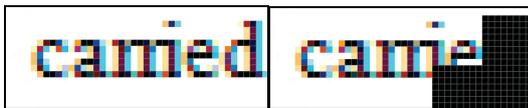


Figure 3. Left: A whole word. Right: Hidden characters due to vision problems.

Taking into consideration that the LVA is not adequate to solve this problem and that it is necessary for the rehabilitation of a person with this particular type of vision problems, it appears that another type of procedure is needed. This procedure consists of preparing the working area for posterior calculations that will give form to the algorithm. Thus, the process is presented in the following steps.

#### 3.1 Divide the working place in two sections.

Taking as reference the middle point of the area that the user cannot see, the division is made so to separate this zone in two sections. After this phase, the algorithm is applied to each one of the zones independently. Thus, both sides, avoiding a saturation of the content in only one side, can observe the hidden content.

Once this is performed, the next step is to calculate the target area.

#### 3.2 Localization of the target area

Next, the working area is calculated for each section; this is the zone that stores the resultant pixels of the fusion between the pixels of the visible and the pixels of the non-visible areas. To perform that, the location of each of the peripheral pixels is identified in a determined section and then multiplied by two the pixel farther of the centre of the division. Thus, the area between the calculated space and the end of the section provides the

working space. The following figure (Figure 4) presents the result of calculating the working area (target area) based on example as appears in Figure 3.

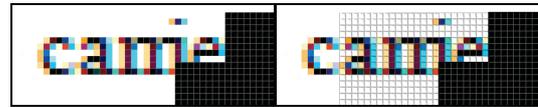


Figure 4. Resulting working area.

### 3.3 Working area pixels' calculation Reading Algorithm.

The number of resulting lines of the working area corresponds to the height of the non-visible area. For each, the pixels from the perimeter to the final of the working area (end of the visible area) are calculated. This is possible as each pixel in a  $p$  position into the line is determined by the mean of the pixels' values which position is  $(p \times 2) y (p \times 2) - 1$  into the same line. As result of these calculations, the contained information is obtained behind the non-visible area that passes to the user's visible area, facilitating reading because the reader can see every character of a specified word. Figure 5 depicts they way the whole word is shown after applying the algorithm. For the pixels outside the previous calculations, the colour is the predominant colour in the working area, in this case, white.

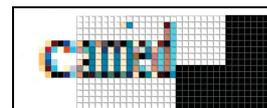


Figure 5. After applying the LVRA.

Thus, a user with vision difficulties is able to read in a much easier way the content of a text file, a web page or any other element formed by characters.

The main core of the algorithm used is expressed in pseudocode as follows:

```
//lpp = list of peripheral pixels
//center = closest pixel to interjection
function LowVisionReadingAlgorithm (ListPixel lpp, Pixel center)
{
  Lines l = GetLines(lpp, center);
  foreach(Line b in l)
  {
    for(int i=0; i<length(b); i++)
    {
      LPixel p = b[i];
      ListPixel lrp = p.getRefPixels();
      int sumR = 0;
      int sumG = 0;
      int sumB = 0;
      foreach(LPixel rp in lrp)
      {
```

```

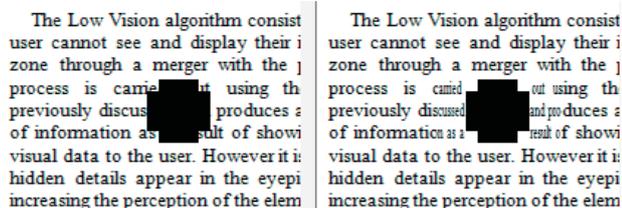
        Color c = rp.getColor();
        sumR = sumR + c.red;
        sumG = sumG + c.green;
        sumB = sumB + c.blue;
    }
    p.setColor(sumR/lenght(1rp),sumG/lenght(1rp),
sumB/lenght(1rp));
    }
}
}

```

The functionalities are:

- *GetLines (ListPixel, Pixel)*. Returns a list of *Lines*, which has in its first position the corresponding peripheral pixel, and in the last position the central pixel. Each pixel will be a *LPixel* object that contains information related with the colour, if inside the working area, its position in relation with the axes and a list of references to the pixels taken to recalculate the original.
- *LPixel.getRefPixels()*. Returns a list of *LPixel* used to calculate the new colour of the original pixel.

Once the algorithm has finished and after obtaining the resulting image, the same procedure is performed for the next part. Thus, at the end of the process the resulting image reveals the hidden information that the reader/user's visual field is non-visible (Figure 6).



**Figure 6.** At left, text without using the LVRA. At right, text that uses the LVRA.

#### 4. CONCLUSIONS

This paper presented the Low Vision Reading Algorithm (LVRA) aiming at helping people with problems related to low vision, associated to visibility losses in specific areas. LVRA supports the reader in performing the reading task easier when using a mobile device. Synthesising two algorithms, LVA and LVRA, forms a global application to use with any content format.

In addition, due to the fact that the algorithm works in the background of the device, it performs without noticing as the user redirects the device's camera to specific text areas, using it as an intermediary between the user and the world to supporting the user in seeing the details around the specific locus.

#### 5. REFERENCES

- [1] Habib M. Fardoun, Lorenzo C. González, Abdulfattah S. Mashat. Rehabilitation Low Vision Algorithm: For People with Central or Multiple Losses of Vision. *7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, 2013. Pages: 339 – 343. E-ISBN: 978-1-936968-80-0.
- [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. Clínica 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.