JEZIK: A Cognitive Translation System Employing a Single, Visible Spectrum Tracking Detector

Davor Bzik

Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Electrical and Computer Engineering Commons

BYU ScholarsArchive Citation


https://scholarsarchive.byu.edu/etd/5924

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
JEZIK: A Cognitive Translation System Employing a Single, Visible Spectrum Tracking Detector

Davor Bzik

A thesis submitted to the faculty of Brigham Young University in partial fulfillment of the requirements for the degree of Master of Science

James K. Archibald, Chair
D. J. Lee
Doran Wilde

Department of Electrical and Computer Engineering
Brigham Young University

June 2016

Copyright © 2016 Davor Bzik
All Rights Reserved
ABSTRACT

JEZIK: A Cognitive Translation System Employing a Single, Visible Spectrum Tracking Detector

Davor Bzik
Department of Electrical and Computer Engineering, BYU
Master of Science

A link between eye movement mechanics and the mental processing associated with text reading has been established in the past. The pausing of an eye gaze on a specific word within a sentence reflects correctness or fluency of a translated text. A cognitive translation system has been built employing a single, inexpensive web camera without the use of infrared illumination. It was shown that the system translates the text, detects rarely occurring and out-of-context words from eye gaze information, and provides solutions in real time while the user is still reading. The solutions are in form of a translation, definition or synonym for the word in question. The only effort required is that of reading.

Keywords: pupil tracking without infrared, web-camera eye tracking, eye gaze estimation, cognitive translation, the Jezik system, assistive translation, inexpensive translation assistant, reading assistant
ACKNOWLEDGEMENTS

My deepest and everlasting gratitude goes to my wife, Edita, the electrical engineer who selflessly took care of our children and provided moral strength for the whole family throughout this journey.

My appreciation and gratitude goes to the BYU's Electrical and Computer Engineering Department that provided significant funding resources enabling me to complete the Master’s program. Special thanks goes to Dr. James K. Archibald, my advisor, whose doors were always open, who patiently but firmly provided the guidance, inspiration and assurance for a successful academic maturation; to Dr. Brian Jeffs who provided support by readily resolving obstacles that entered the picture; to the committee members, Dr. D. J. Lee and Dr. Doran Wilde, for their guidance and timely humor; and to the department's staff: Kris, Joe, Don, Mark and Janalyn for extraordinary service and kindness.

My appreciation and love goes to my mother and father – something about the potatoes and digging deeper; to our dear friends Sandy and Ali, Holly and family, Linda and Patrick, Jerianne Hanks, Dustin Hill and his family, Cammie and Creed Archibald, the Lewis family and many other friends who take pride in our family's accomplishments.

Finally, one man watchfully stood behind me whose profound advice and encouragement were unprecedented – Dr. Douglas Čabrajec . . . many thanks to him and his good wife, Ada, for wonderful family moments.
# TABLE OF CONTENTS

TITLE PAGE ................................................................................................................................... i

ABSTRACT .................................................................................................................................... ii

ACKNOWLEDGEMENTS ........................................................................................................... iii

TABLE OF CONTENTS ............................................................................................................... iv

LIST OF FIGURES ....................................................................................................................... vi

LIST OF TABLES ........................................................................................................................ viii

1 INTRODUCTION ................................................................................................................... 1
  1.1 History .............................................................................................................................. 1
  1.2 Why a cognitive translation framework ........................................................................... 2
  1.3 Research objective ............................................................................................................ 3
  1.4 Delimitations .................................................................................................................... 3

2 BACKGROUND AND PREVIOUS WORK ......................................................................... 5
  2.1 Camera parameters, CMOS sensors and internal processing algorithms ......................... 5
  2.2 Eye gaze estimation .......................................................................................................... 9
  2.3 Cognitive process in reading .......................................................................................... 10

3 THE JEZIK SYSTEM ........................................................................................................... 13
  3.1 The aim of the research .................................................................................................. 13
  3.2 Scope .............................................................................................................................. 16
  3.3 Approach ...................................................................................................................... 16
  3.4 Motivation ...................................................................................................................... 17

4 EXPERIMENTAL EQUIPMENT AND PROCEDURE ...................................................... 19
  4.1 Environment (lighting conditions) ................................................................................... 19
  4.2 Camera ........................................................................................................................... 21
  4.3 Set-up ............................................................................................................................. 22
  4.4 Camera calibration and parameters ................................................................................ 22

5 USER INTERFACE AND ALGORITHMS ............................................................................. 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Software organization and design</td>
<td>27</td>
</tr>
<tr>
<td>5.2</td>
<td>Sequential logic</td>
<td>29</td>
</tr>
<tr>
<td>5.3</td>
<td>Camera parameters</td>
<td>33</td>
</tr>
<tr>
<td>5.4</td>
<td>Region of interest and image preparation</td>
<td>39</td>
</tr>
<tr>
<td>5.5</td>
<td>Pupil Detection</td>
<td>42</td>
</tr>
<tr>
<td>5.6</td>
<td>Eye Detection</td>
<td>46</td>
</tr>
<tr>
<td>5.7</td>
<td>Head detection and pose estimation</td>
<td>48</td>
</tr>
<tr>
<td>5.8</td>
<td>Eye gaze estimation</td>
<td>49</td>
</tr>
<tr>
<td>5.9</td>
<td>Translation module</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>RESULTS</td>
<td>55</td>
</tr>
<tr>
<td>6.1</td>
<td>Camera resolution dependency</td>
<td>55</td>
</tr>
<tr>
<td>6.2</td>
<td>Pupil detection precision</td>
<td>55</td>
</tr>
<tr>
<td>6.3</td>
<td>Head pose and eyes detection precision</td>
<td>56</td>
</tr>
<tr>
<td>6.4</td>
<td>Eye gaze precision and speed</td>
<td>56</td>
</tr>
<tr>
<td>6.5</td>
<td>Comparison with commercially available tracking systems</td>
<td>59</td>
</tr>
<tr>
<td>6.6</td>
<td>Different lighting conditions</td>
<td>59</td>
</tr>
<tr>
<td>6.7</td>
<td>Accuracy of the cognitive load measurement</td>
<td>60</td>
</tr>
<tr>
<td>7</td>
<td>CONCLUSION AND FUTURE WORK</td>
<td>63</td>
</tr>
<tr>
<td>7.1</td>
<td>What was done</td>
<td>63</td>
</tr>
<tr>
<td>7.2</td>
<td>Future research</td>
<td>63</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Improving gaze detection</td>
<td>64</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Extending the translation module</td>
<td>66</td>
</tr>
<tr>
<td>8</td>
<td>BIBLIOGRAPHY</td>
<td>69</td>
</tr>
<tr>
<td>APPENDIX A</td>
<td>GLOSSARY OF TERMS</td>
<td>73</td>
</tr>
<tr>
<td>APPENDIX B</td>
<td>TEST SENTENCES</td>
<td>75</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1-1: SDL Trados Studio translation software interface. Image acquired from [1] through Bing search engine under "Free to share and use" license............................................................. 1

Figure 2-1: Comparison of rolling shutter and global shutter side-effects. Image acquired from [16] through Bing search engine under "Free to share and use" license. .................................................. 8

Figure 3-1: The Cognitive translation system employing a single, visible spectrum tracking detector................................................................................................................................. 14

Figure 4-1: Placement of lighting sources in a typical translation environment ........................................... 21

Figure 4-2a-c: The effects of changing camera gain for a single grayscale test pattern shown in Figure A. Figure extracted from [10]. ........................................................................................... 23

Figure 4-3: Camera's radial distortion due to lens limitation (courtesy of Anna G. from Edmund Optics Inc.) ............................................................................................................................. 25

Figure 5-1: Class dependencies for the Jezik system......................................................................................... 28

Figure 5-2a: Sequential processing logic of the Jezik system (part A). .......................................................... 30

Figure 5-2b: Sequential processing logic of the Jezik system (part B). .......................................................... 31

Figure 5-3: Original image with factory default camera parameters.............................................................. 37

Figure 5-4: Original image after adjusting the camera parameters for image segmentation......................... 38

Figure 5-5: The grayscale image of a red channel after the camera parameters are adjusted and red channel extracted from R'G'B'. ............................................................................................... 38

Figure 5-6a: Starting position for the ROI and the pixel. The pixel occupies half of two neighboring pixels .......................................................................................................................... 39

Figure 5-6b: The pixel has been moved to the left by a half pixel. The ROI is moved accordingly so that the pixel is always positioned in the center................................................................. 40

Figure 5-6c: The ROI needs to be rounded for the next iteration. If the rounding causes the ROI to not be centered around the pixel, it is rounded up to the nearest whole pixel................................. 40

Figure 5-6d: The pixel moves by a half pixel to the right, once again occupying two neighboring pixels. The ROI follows............................................................................................................... 40

Figure 5-6e: Once again the ROI is rounded, but this time it is rounded down, since it can be centered around the pixel............................................................................................................. 41
Figure 5-7: The pseudo-code for detection and tracking of pupils

Figure 5-8a: Left pupil ROI without up-sampling

Figure 5-8b: Left pupil ROI after thresholding

Figure 5-8c: Left pupil ROI after applying morphological open operator with 5x5 disc kernel

Figure 5-9a: Left pupil ROI with 3x up-sampling

Figure 5-9b: Left pupil ROI after thresholding

Figure 5-9c: Left pupil ROI after applying morphological erode and close operator with 5x5 disc kernel

Figure 5-10: The pseudo-code for detection and tracking of eye center

Figure 5-11a: Left eyebrow ROI without up-sampling

Figure 5-11b: Left eyebrow ROI after thresholding

Figure 5-11c: Left eyebrow ROI after applying two morphological open operators with 5x5 disc kernel

Figure 5-12a: Nose ROI without up-sampling

Figure 5-12b: Nose ROI after thresholding

Figure 5-12c: Nose ROI after applying morphological open operator with 5x5 disc kernel

Figure 5-13: Gaze estimation focal points

Figure 5-14: Translation interface in Wordfast, typical of an assistive translation system. Image acquired from [31] through Bing search engine under "Free to share and use" license

Figure 5-15: Visual display of the translation screen in the Jezik system

Figure 6-1: The Jezik system in use

Figure 7-1: Final results showing the basis for the geometric head-pose model

Figure 7-2: Outline of the calculation steps to determine the head pose detection (Figure extracted from [33])
LIST OF TABLES

Table 4-1: Lighting source placement and intensities ................................................................. 20

Table 5-1: Logitech C920 camera settings mapping to OpenCV functions. Color Intensity* is denoted as Saturation. ........................................................................................................................................ 33

Table 5-2: Irregularities in step size in the Logitech C920 camera .............................................. 34

Table 6-1: Individual algorithm processing time ........................................................................... 57

Table 6-2: Total fixations (adjusted) for various LF word scenarios ............................................. 61
1 INTRODUCTION

1.1 History

For over 30 years human translation processes have been studied to establish a correlation between eye movement mechanics and the mental processing associated with text reading. In the past 20 years translators have relied upon methodologies such as translation memory (a database that stores segments of text that have previously been translated), terminology managers, shared...
dictionaries etc, all of which are incorporated into computer-assisted translation solutions such as *Wordfast* [2] or *SDL Trados* [3] (see Figure 1-1). While these tools can generally help the translator improve consistency and the quality of translations, the human factor is still present with publication deadlines, attempts to preserve meaning, and the use of idioms and metaphors, all potentially hindering the creative process and yielding less than satisfying results. As Figure 1-1 shows, the latest editions of SDL Trados Studio add a layer of complexity to the user interface defocusing the translator’s attention away from the translation text.

Previous studies suggest that the link between the reading process and eye movement exists and that it needs to be further established [4]. With the increase of computing power, various approaches and algorithms have been devised to track eye movements, furthering research into the reading process.

### 1.2 Why a cognitive translation framework

Automated language translation techniques and programs generally suffer from awkwardness in the final language translation due to inaccurate placement of words and awkward uses of prepositions and articles. For a native speaker, the eyes immediately become fixed on the problems in the translation from another language. To improve the translation quality and speed, feedback from the translation tool based upon the gaze of the eyes and the location of the problems offers great promise in improving translation. Eye tracking is an active area of research for evaluating the cognitive effort involved in reading text [5]. For a translation tool, the pausing of the eyes to gaze on a specific phrase reflects the correctness or fluency of a translated text. The only effort required is that of reading. For a tool helping developing readers,
automated feedback based on eye gaze could assist in the comprehension process by resolving the lexical meaning of sentences [6].

1.3 Research objective

The goal of this work is to determine if an inexpensive translation system can be built using a single, widely available, low-priced web camera that allows for real-time assisted translation and assistive reading, without employing any additional devices that could cause distraction or discomfort, such as infrared illumination, head-mounted cameras or a bite bar [7].

As part of this research, a significant emphasis was placed on the ability to operate the tracking system and the translation system in real-time. Many previous systems have been designed to operate offline, using commodity computers, and hence cannot support the required operations in real-time. In contrast, the system proposed in this thesis was designed for real-time operations. This system includes a graphical user interface (GUI) for cognitive translation and feedback to re-evaluate the output based on the user's eye movements. An essential component of this system is the presentation of real-time solutions to the user.

While systems described in the literature have used expensive IR cameras, off-line processing, head-mounted trackers and head constraints, this research will consider a system using a single visible-spectrum camera without additional equipment or constraints.

1.4 Delimitations

The goal of this research is to show the feasibility of an automated translation system as a whole and thus the emphasis is not on optimizing individual modules such as facial feature detection, tracking, pose estimation, eye gaze estimation or cognitive load mapping. The current
system is a starting point, and certainly not a commercially viable product. The interface proposed in this research is suited for adding new functionalities and algorithms that would unleash the full potential of automated translation. One example is the algorithmically inexpensive facial feature tracker for tracking eyes, which could certainly employ more sophisticated algorithms. By design, the algorithms proposed in this research are fast, robust and utilize only a fraction of the available CPU capacity on a standard laptop. Improved precision can be achieved using a more complex algorithm for eye center localization, for example.

The testing environment resembles a common translator's environment and the system is calibrated for common lighting conditions. Any drastic alterations might require re-calibration of the system, which is done manually, but could be automated in the future.

Three distinct types of readers exist:

- fast linear readers
- slow linear readers
- topic structure processors

Linear readers read words in order while topic structure processors glance at words and phrases to quickly scan text [8]. This research targets individuals who read words linearly.

For the prototype system described in this work, a test sentence database and dictionary was formed from a set of hand-selected sentences and control words. Enhanced revisions of this system would naturally be linked with a full-scale commercial translation memory database and external dictionaries to increase its usefulness in a more realistic setting.
2 BACKGROUND AND PREVIOUS WORK

2.1 Camera parameters, CMOS sensors and internal processing algorithms

The prevalent technology for the photosensitive sensory element in a web camera is complementary metal-oxide semiconductor (CMOS) due to the low production cost and acceptable quality for many consumer applications. All photosensitive elements based on silicon are inherently sensitive to a much wider spectrum than the human eye can see. The wavelength range usually extends between 350nm to 1050nm; however, an infrared (IR) cut-off filter (700nm - 1100nm) is typically placed in front of the lenses to lower the noise in the visible spectrum.

Although a variety of web cameras is available on the market today, they all use similar parameters to adjust the image quality under different lightning environments. Web camera manufacturers have a variety of different color spaces at their convenience to choose from, such as $\text{RGB}$, $\text{CMY(K)}$, $\text{HSL}$, $\text{YIQ}$, $\text{YUV}$, $\text{YCbr}$, $\text{YCC}$, $\text{CIELuv}$, $\text{CIELab}$, etc [9]. Using capabilities supported in the device firmware, the user can change several different parameters to adjust the image quality.

Logitech uses 6 key parameters in most of their camera models: $\text{exposure}$, $\text{gain}$, $\text{brightness}$, $\text{contrast}$, $\text{color intensity}$ and $\text{white balance}$. Other manufacturers might use a combination of $\text{hue}$, $\text{saturation}$, $\text{gamma}$ and $\text{tint}$ that replace the $\text{gain}$, $\text{color intensity}$ or $\text{white balance}$. The research objective for automated translation is not concerned with the color space.
used, but rather how to achieve the desired image filtering by adjusting the sensor parameters. For the purposes of this thesis, *zoom, orientation* and *focus* will be combined with the 6 parameters used by Logitech in their control interface. Specifically, the parameters used are the following:

*Exposure* – determines how long the photo-reactive element is exposed to the light. The result of underexposing is an image with a low signal-to-noise ratio. Overexposure will lead to saturated pixels with no detectable image detail [10]. Another side effect of high exposure is motion blur – the apparent streaking of rapidly moving objects in a still image – which increases with the camera's exposure time.

*Gain* – the amount a signal increases while going through an amplifier. It varies the output value from the camera for a given change in input. It is expressed as a ratio of the number of electrons from the CMOS chip that are converted into analog-to-digital units. Increasing the camera gain effectively means reducing the number of electrons required to produce one analog to digital unit (ADU) [11]. The effect this has on the captured image is to multiply all pixel values by the same value. For example, if two pixels initially have gray values of 40 and 50, thus having a difference of 10, increasing the gain from 1 to 3 will result in new values of the pixels as 120 and 150, with a difference of 30. Caution must be employed to ensure that the significant image features and precision are not lost due to incorrect gain settings.

*Brightness* – a measure of intensity after the image has been acquired. Brightness adjustment is a linear operation [12] in which a constant is added to or subtracted from the pixel values.
Contrast – a measure of the difference in brightness between light and dark areas in a scene [13]. For an image to appear well defined, the black details need to appear black and the white details must appear white. The more the black and white information trend into the intermediate grays, the lower the contrast at that frequency. The greater the difference in intensity between a light and dark line, the better the contrast. Illumination plays a key role in determining the resulting image contrast [14].

Color intensity – defines a range from pure color to gray at a constant lightness level. It is expressed in percentages. From a perceptional point of view, it influences the grade of purity or vividness of a color. Images with low color intensity appear dull or washed out [15].

White Balance – a colorimetric parameter associated with light sources that relate to apparent visual color. Cameras are calibrated to recognize the color white as white so that all colors appear normal. It is expressed in degrees Kelvin. Higher numbers (> 6000 K) render warmer, red hues, while lower numbers (3000 K) will render colder, blue hues.

Zoom – a function of a digital camera used to make the image seem more close-up, by cropping and enlarging. In contrast to the optical zoom present in consumer cameras where lenses control the image size, digital zoom uses up-sampling algorithms to render the close-up image. The quality of the zoomed image depends on the algorithms employed in the camera.

Orientation – a function of a digital camera used to capture the image in a normal orientation or a mirrored orientation.

Focus – a function of adjusting the range of distances within which the objects will appear clearly on a captured image. Low-end web cameras have fixed focus, while the mid- and high-end cameras use a small, electronically controlled lens that allows the adjustment of focus.
For the work described in this thesis, it was important to explore the parameter settings that affected the ability to segment each image to show only features of interest.

Integrated cameras often use an electronic rolling shutter which exposes, samples and reads out pixels sequentially, as opposed to a global shutter where all the pixels are exposed and sampled simultaneously. In the rolling shutter case, each line of the image is sampled at a slightly different time, thus introducing distortion in images of moving objects at high speeds. Figure 2-1 illustrates the difference in distortion between rolling and global shutter. The images are of a three blade propeller with all blades of equal size. While the global shutter version has minimal geometrical distortion, it has more blur.

![Rolling Shutter vs. Global Shutter](image)

Figure 2-1: Comparison of rolling shutter and global shutter side-effects. Image acquired from [16] through Bing search engine under "Free to share and use" license.

The frame rate (fps) refers to the number of full frames captured in a second. The shutter speed corresponds to the inverse of the exposure time of the sensor. The exposure time controls the amount of incident light collected by the sensor. Camera blooming (an overflow of charge
from neighboring pixels) near over-exposure can be controlled by decreasing illumination, or by increasing the shutter speed, which decreases the exposure time.

2.2 Eye gaze estimation

When an active, near-infrared lighting source is placed close to the optical axis of a camera, the pupils look unnaturally bright because of reflections on the fundus – the interior lining of the eyeball. Anyone familiar with photography has noted this red-eye effect. The pupillary light reflection is very weak under direct near-infrared illumination. This is beneficial for eye detection based on the bright pupil effect but raises safety concerns when using infrared radiation.

Long-term exposure to infrared radiation is hazardous to the eyes. Since a translator's everyday job requires 6-8 hours of exposure to a system, even low-level infrared radiation accumulatively becomes hazardous and can lead to a thermal injury to the cornea and the lens of the eye [17], [18]. Also, infrared signals are easily spoiled by the presence of other infrared sources, such as sunlight or wireless consumer electronics.

Generally, high illumination levels allow faster shutter speeds, which reduce motion artifacts, and smaller apertures, which give larger depths of field. Also, the signal-to-noise ratio of an imaging sensor generally increases with higher illumination levels [19].

Below are outlined some of the recent eye tracking and eye gaze estimation approaches:

Ince and Kim [20] achieve a performance rate of 15fps for small input resolution images of 160x120 which is insufficient for tracking saccadic (rapidly irregular) eye movement. Visual angle accuracy is around 2 degrees.
Timm and Barth [21] describe a feature-based method that achieves a position accuracy of 82.5% for pupil localization, but the authors do not mention real-time performance of the algorithm.

Reichle et al. [22] present a system consisting of two cameras: one narrow view and one wide view, employing IR illumination. Although the precision is enhanced, it performs pupil detection at only 19 fps.

Heyman et al. [23] show a robust method for eye gaze estimation; however, the authors state that the optimized version achieves a tracking speed of 64 ms per frame or 15.6 fps.

Commercial eye trackers such as Tobii PCEye Go [24] are state-of-the-art in versatility and precision, with an entry-level model at a price 100 times that of a web camera. Due to the expense of these systems, they are generally limited to use in research as a tool for evaluating psychological-cognitive concepts or as assistive tools for people with disabilities. This leads to the key question addressed in this thesis: can an inexpensive eye tracker be developed with sufficient accuracy to be used for cognitive research?

2.3 **Cognitive process in reading**

When visually processing words, eyes do not move continually along a line of text but make short rapid movements (saccades) intermingled with short stops (fixations) and backward-directed eye movement (regressions) [25]. The distance the eye moves with each saccade varies between one and 20 characters, with an average fixation duration of 200ms [4]. Slower readers have longer fixation durations and make shorter saccades as well as more regressions (re-readings) than faster readers [7].
Previous studies suggest that the cognitive processes used to recognize frequently used words and seldom used words may vary with the amount of contextual constraint and the level of reading achievement. Words that appear frequently in written text are read more quickly (i.e., shorter eye fixations) than are seldom used words. Similarly, words that are predictable from prior context are read more quickly and are less likely to cause the reader to pause than are unexpected words [7].

A simple example of a person learning English who might say "I am going to bank" is readily understood by English speakers. However, native English speakers would immediately focus their eyes on the written text where the article "the" should be inserted to produce the sentence "I am going to the bank." The purpose of this work is to identify through eye tracking where the cognitive issue is located and provide corrective alternatives from a translation dictionary. Various examples can be found in [6]. The overall goal of this work is to show the feasibility of this general approach and to explore features desirable in a viable commercial product.
3 THE JEZIK SYSTEM

3.1 The aim of the research

Even with the increasing ability of modern processing power and clever algorithms to translate for us, any qualitative translation is still done manually by humans. As a result, it has been postulated that human cognition could be enhanced by employing high-speed eye tracking. For an average translator it would seem unreasonable to invest approximately $3000 [26] into high-end tracking equipment and proprietary software. The Jezik system (the name Jezik means "language" in Croatian) proposed in this thesis is a cognitive translation system employing a single, visible spectrum tracking detector. The aim of this thesis and research is to find whether a proof-of-concept system can be built using a single inexpensive web camera, without infrared lighting, that would perform tracking, gaze estimation and cognitive load detection in real-time. If the investment for such a system were limited to the cost of a $50 web camera and a modest monthly subscription fee (for access to an online translation database), a translator could easily afford to use such a system.

Figure 3-1 shows a proposed system for measuring the cognitive load on a human brain using statistical data extraction from an eye tracking component feature (the Personal Eye Tracker). The Statistical Model in the figure is built around a machine-learning algorithm that quantifies eye tracking parameters into a measure of cognitive load for a unit of interest (a word, sentence, or paragraph). The data is then fed into the Real-time Suggestor and Efficiency Monitor.
components from which a variety of translation and assistive learning-related applications are derived:

- **Whisperer** – a presentation layer that draws on-screen solutions for the user and tracks the user's response in order to tailor the length of the presentation and to determine when to start a new presentation.

- **Efficiency alerts** – tracking of personal work progress and providing notifications on efficiency.
• *Supervision* – combines efficiency data from multiple users to provide an overall report on how established translation steps impact speed and quality of translation.

• *Personal training* – self-assessing tool that combines efficiency data from a user and provides feedback on speed and quality for units of paragraphs as well as overall steps in a translation project.

• *Adaptive learning* – tracks user's learning progress in comprehension and adapts the study material accordingly. Possible uses are early childhood development research, reading assistants, study assistants, etc.

• *Real-time Suggestor* – a collection of Application Programming Interfaces (API) providing a bridge between the *Statistical Memory Translation* (SMT) system – such as Google Translate – or existing translation memories, and the *Whisperer* application.

   It is assumed that there exists a high correlation between eye movement patterns and cognitive discomfort with what is being read – this is the basis of this research.

   Ideally, the *Statistical model* should provide answers to the following questions:

   - Is the user tired or continually distracted?
   - Which word is causing the cognitive load to be above a pre-determined threshold?
   - Which portion of a sentence is causing the cognitive load to be above a pre-determined threshold?
   - Which sentence is causing the cognitive load to be above a pre-determined threshold?
   - Which paragraph is causing the cognitive load to be above a pre-determined threshold?
From these measurements we seek to determine typical characteristics of cognitive loads in sentences or paragraphs to identify useful descriptors.

3.2 Scope

The goal of this research is to show that such a cognitive translation system is possible. As such, it is necessary to develop minimum requirements for the Personal Eye Tracker, Statistical Model and Real-time Suggestor to accommodate the Whisperer application in order to show that the system is feasible. A significant concern is whether or not the system can operate in real-time using standard commodity computing components and accessories. It will be shown that such is indeed both possible and feasible.

3.3 Approach

The camera's parameters were researched and adjusted to employ the camera's internal algorithms for on-camera image processing. By pre-processing the image internally in the camera, one avoids subsequent post-processing. The motivation is to increase overall processing speed because once the image is captured and transferred into computer memory, the image processing becomes the critical constraint. After determining the best adjustments for the camera, a GUI was developed, which enables the user to perform the calibration and comfortably interact with the system. Further, a series of reading assignments comprised of sentences containing control words were performed to test the efficiency of the system. These tests will be discussed in the results chapter.
3.4 Motivation

The Church of Jesus Christ of Latter-day Saints employs translators for over 120 languages [27] to meet deadlines for their semi-annual world-wide broadcasts. In addition, teaching materials, talks, training sessions and hymnbooks are translated on a regular basis. In translation projects, work is usually divided into four assignments:

- translation
- content reviewing
- language reviewing
- proofreading

where each subsequent assignment is dependent upon the previous. In critical projects each assignment is given to a different translator. Large projects tend to be divided into smaller portions, each assigned to a group of translators.

Having observed the magnitude of this work as an active participant in the process, it became apparent to me that technology would greatly assist in this enormous task. It is my desire to provide a system (like the Jezik system) to the many thousands of translators.
4 EXPERIMENTAL EQUIPMENT AND PROCEDURE

4.1 Environment (lighting conditions)

The suitability of the lighting environment under which the tracking algorithm performs as intended is determined by two key factors:

- the frontal uniformity of the light source, and
- the ratio of frontal vs. background light intensity.

Frontal uniformity is achieved when a light source is being projected onto the user's face from the front, generally aligned with the user's head both horizontally and vertically. Large horizontal offsets of a light source create shadows on the user's face which might lead to detection with less precision (Figure 4-1). Vertical offsets such as the light source being placed above the user's head so that the face is still illuminated while not blinding the eyes are desirable. Smaller vertical offsets, such as light sources being placed at the height of the user's eyes create corneal reflections which can affect the tracker's precision. The ratio of

\[
\frac{\text{frontal light intensity}}{\text{background light intensity}}
\]

has been determined by measuring the light intensities 0.5 meters in front of the user and 0.5 meters behind the user with a color and angle corrected photometer. The measurements are given in Table 4-1.
From experimentation, it was found that it is desirable for the frontal light source to be turned on, and that the desired ratio of frontal and background light intensity was found to be

\[
\frac{90}{62} = 1.45.
\]

Lowering the ratio in subsequent experiments by approximately 20% did not negatively impact the detection algorithm. It is possible that even lower ratios may work, but the exposure setting would need to be adjusted to compensate.

The background light source was a 4100 Kelvin fluorescent lamp and the frontal was 3500 Kelvin. The system was also tested with LED (3000 Kelvin) and incandescent (2000 Kelvin) lights, as well as direct sunlight as frontal light sources and it performed well with all of them. However, when using the LED light, the intrinsic noise in the image was somewhat reduced.

It is also worth mentioning the reference lighting conditions under which the detection algorithm achieves best results – that is by using only a frontal light source placed above the user's head and indicated as “Optimal” in Figure 4-1. Such placement is not uncommon for translators working late hours with a desktop lamp on. In some cases, the brightness of the monitor screen is great enough to serve as a light source in a dark room.

<table>
<thead>
<tr>
<th>PLACEMENT OF PHOTOMETER</th>
<th>LIGHT INTENSITY (foot-candles)</th>
<th>LIGHT INTENSITY (lux) [28]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Behind the user's head. Frontal light source off.</td>
<td>60</td>
<td>645</td>
</tr>
<tr>
<td>In front of the user's head. Frontal light source off.</td>
<td>38</td>
<td>409</td>
</tr>
<tr>
<td>In front of the user's head. Frontal light source on.</td>
<td>90</td>
<td>969</td>
</tr>
<tr>
<td>Behind the user's head. Frontal light source on.</td>
<td>62</td>
<td>667</td>
</tr>
</tbody>
</table>
4.2 Camera

The camera used is a Logitech C920 costing $40. The driver version is 13.51.823.0, obtained from the manufacturer's website [29]. No modifications have been made to the camera, i.e. the infrared filter is not removed. The camera is capable of delivering 30 fps at 1920x1080 resolution; however, USB2 limits the transfer bandwidth to 15 fps at that resolution, so the lower 1280x720 resolution is used in this work. Full HD resolution can be employed at 30fps if H.264 encoding is employed, which is supported by the camera, internally compressing the image.
stream. Other cameras that were used for testing included the Microsoft LifeCam Studio costing $50 and the Logitech C270 costing $20. Both are suitable for the task, but the C920 was selected because it uses a low-noise CMOS sensor.

For eye tracking purposes in cognitive load detection, the higher the camera frame-rate, the better the precision in measuring the detected saccadic eye movements. Saccadic eye movement has been measured to be in the 20 ms range [30] which corresponds to a frame-rate of a 50 fps, so it is crucial to employ the web camera's full capabilities.

It should be noted that the Logitech's stated frame-rate varies somewhat around the specified rate. Specifically, in the Logitech camera, a specification of 30 fps resulted in frame-rates that varied from 28fps to as much as 45fps; however, when the frame-rate approaches 45fps, the camera notes this variation and resets the rate to 30fps. This anomaly was been observed through experiments.

4.3 Set-up

The camera is strategically positioned slightly below the user's chin, i.e. below the monitor screen, so that the pupils and nostrils are visible at all times during head movement, assuming a normal head position. The distance between the user's head and the camera is approximately 50cm. The user is seated in a comfortable position and the camera is moved so that the optical center is approximately aligned with the center of the head.

4.4 Camera calibration and parameters

All automatic camera adjustments, such as auto white balance, auto focus, proprietary options and any other settings that might dynamically affect the image quality need to be
disabled. The goal of modifying the camera parameters was to change the color mapping which allowed detection of the pupils, nostrils and eye brows. To do this, the camera parameters were adjusted as will be explained in Chapter 5, so that the R channel of the RGB image allows the detection of the location of the pupils and nostrils. Most modern cameras are equipped with a "red-eye" mode to remove the red pupil coloring, while in our case we are trying to emphasize the location of the "red-eye." Serendipitously, the procedure which emphasizes the location of the pupil also emphasizes the location of the nostrils.

It was determined experimentally, as will be shown in Chapter 5, that the eyebrows were best located using the grayscale image obtained from the mapped RGB image.
Adjusting the overall camera gain allows more light to fall on the image sensor and thus improves image quality in the darker regions. Figure 4-2a-c shows the effects of changing camera gain on a grayscale test pattern. The y-axis shows gray level ranging from 0 to 255, and the x-axis shows position across the image.

The effects of changing camera gain for a single grayscale test pattern are shown in Figure 4-2a. The image shows the digitized camera output using all of the 8-bit camera output. Figure 4-2b shows the effect of increasing the camera gain where the output maps the brightest input values into a single 8-bit output value of 255, thereby effectively reducing the range. Figure 4-2c shows the effect of decreasing the camera gain in which the digital output values above 128 are never used, which also reduces the range.

It is desirable to adjust the camera gain in order to achieve maximum separation between the digitized values of the facial features (i.e., pupils, nostrils and eyebrows) and the rest of the image.

The image captured by a camera can be processed and enhanced using image processing software. However, greater precision can be obtained by adjusting the internal camera parameters and using the internal camera software to filter out the desired features. Properly adjusted, the final image will allow the accurate detection of the pupil location and also reduce the processing load for the Jezik system.

The camera's radial distortion due to the lens limitation is not addressed in the Jezik system, since feature detection occurs in close proximity to the optical center of the camera, where the distortion is minimal. Figure 4-3 shows negative or barrel distortion in a camera lens system, where pixel information is generally geometrically misplaced. Distortion is calculated by
relating the Actual Distance (AD) and the Predicted Distance (PD). White circles represent the actual pixel scene locations, whereas the black pixels represent the apparent locations in the distorted image.
5 USER INTERFACE AND ALGORITHMS

5.1 Software organization and design

In writing the code for this research, one of the major goals was to keep the code readable and the interface simple in order to provide a platform for future research. This was achieved by dividing the system into modules equal to a class (see Figure 5-1). This way, any part of a system can be upgraded with another module while preserving the logic without affecting the rest of the system. Naming of variables and methods adheres to the OpenCV standard interface:

- variable names are concise, descriptive and consistent throughout the program, such as `iLeyeBinary` denoting the binary representation of an image for the left eye.
- function names are descriptive to reflect the processing behind the name, such as `cleanBinaryImageStage1()` or `findEyesCenters()`.
- defensive programming is used whenever possible; objects passed to functions use a `const` reference so the function does not modify the object.
- variables are always initialized and denoted as input, output or input/output variables, such as `const Mat &frameINleft`.
- the code is structured for increased visibility and readability.
- all constants and adjustable parameters are found in the header files of each module or in the implementation file of each class.
Figure 5-1 shows class dependencies in the *Jezik* system. For example, the *FastTracker* class combines communication from multiple classes responsible for tracking of each individual feature.
The *FastTracker* class and the classes it depends on comprise the *Personal Eye Tracker* component from Figure 3-1, while the *WordFinder* and the *MainWindow* classes jointly represent the *Statistical Model, Real-time Suggestor* and the *Whisperer* components.

### 5.2 Sequential logic

Figure 5.2a-b outlines the sequential processing logic of the system as a whole. Those steps outlined by the dashed lines under the MOUTH ROI are part of a future *Jezik* system, which might include mouth detection. The first block of Figure 5.2a adjusts the camera parameters resulting in 3-Domain Color Separation (3-DCS). 3-DCS in essence maps the whole RGB spectrum into the three domains:

- R'
- G'
- B'.

The R' channel contains a narrow range of red and near-red colors. These are displayed as a grayscale image (see Figure 5-5 for example) and used for detection of the pupils, nostrils and mouth. The G' channel and the B' channel are similar in that they accentuate the black and dark colors while suppressing the white and lighter colors. Other than converting the RGB image to a grayscale image, they are not used in this work. A second image is converted into R'G'B' (see Figure 5-4) which is subsequently converted into grayscale for detection of the eyebrows. Section 5.3 describes in detail how this is achieved. The color format used for storing images is OpenCV BGR (Blue, Green, Red), an out-of-order RGB.

There is some degree of similarity between the stages of feature detection so only one will be described here. For nose detection, the region of interest (ROI) is formed at a default location.
Figure 5-2a: Sequential processing logic of the Jezik system (part A)
Figure 5-2b: Sequential processing logic of the Jezik system (part B)
around the optical center of the camera. Once detected, the tracking of the feature is based on previously known positions. The image is pre-processed or blurred in this case, up-sampled to a higher resolution and binarized (converted to an image with only black and white). Up-sampling to about 3, 5 or 7 times the original resolution allows for better image reconstruction in the two cleaning stages where the ellipticity of the object is reconstructed for more precise detection of the feature's center. The cleaning stages can also reduce noise in the image. Then the center detection algorithm is used, after which a linear shift is applied to linearize the distance which the nose travels as the head rotates. From that output the head pose estimation is calculated. The head pose, eye locations and pupil locations are jointly used to compute an eye gaze estimation which is coupled with the Statistical Machine Translator (SMT) – a block comparable to Google Translator – to track the user's eye movement while reading the translated text. Based on tracking information, the cognitive load detection step estimates which word causes confusion in the user’s mind and provides a solution – in real-time – in the form of:

1) a translation of the word in question,

2) a synonym, or

3) a definition

depending on whether the task is translating or reading. The translation is acquired from an external dictionary, which in the case of Jezik system is a hand-composed mini-dictionary. The following sections describe the individual algorithms in more detail.

The algorithmic stages for mouth detection (see Figure 5-2a on the right side) are shown in dotted boxes because they are not used in this research.
5.3 Camera parameters

Adjusting the camera parameters correctly is crucial for a good detection of facial features and this was an important part of the experimental work in this thesis. As noted earlier, the Logitech C920 webcam offers 8 parameters of interest: Exposure, Gain, Brightness, Contrast, Color Intensity, White Balance, Zoom and Focus (see Table 5-1). The additional 3 parameters involved in processing are:

- **Right Light** – unique to Logitech only,
- **Image orientation** – which needs to be set to Mirrored, and
- **Auto focus** which needs to be turned off in the Logitech's Webcam properties dialog.

Other parameters are programmatically accessible though the OpenCV interface. In Table 5-1, a mapping is shown of the Logitech C920 functions available through OpenCV.

Table 5-1: Logitech C920 camera settings mapping to OpenCV functions. Color Intensity* is denoted as Saturation.

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>MIN VALUE</th>
<th>MAX VALUE</th>
<th>STEP</th>
<th>NUMBER OF STEPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exposure</td>
<td>-1</td>
<td>-7</td>
<td>1 - 2 (irregular)</td>
<td>14</td>
</tr>
<tr>
<td>Gain</td>
<td>0</td>
<td>255</td>
<td>3 - 5 (irregular)</td>
<td>51 &lt; and &lt; 85</td>
</tr>
<tr>
<td>Brightness</td>
<td>0</td>
<td>255</td>
<td>2</td>
<td>127 + 1</td>
</tr>
<tr>
<td>Contrast</td>
<td>0</td>
<td>255</td>
<td>4</td>
<td>63 + 1</td>
</tr>
<tr>
<td>*Color Intensity</td>
<td>0</td>
<td>255</td>
<td>1</td>
<td>255</td>
</tr>
<tr>
<td>White Balance</td>
<td>2000</td>
<td>6500</td>
<td>45</td>
<td>100</td>
</tr>
<tr>
<td>Zoom</td>
<td>100</td>
<td>180</td>
<td>20 - 40 (irregular)</td>
<td>16</td>
</tr>
<tr>
<td>Focus</td>
<td>0</td>
<td>250</td>
<td>5</td>
<td>50</td>
</tr>
</tbody>
</table>
Some parameters unfortunately do not have a perfect mapping through OpenCV. Exposure and Gain have irregular step increments, and for Exposure and Zoom not all values are accessible programmatically, although they are accessible through Logitech's property dialog. These irregularities are related to an incompatible OpenCV interface and should be taken into account when making the adjustments. Some webcam manufacturers provide an API for the products, but one was not located for the Logitech C920 camera. However, a very useful application called USB device viewer (UVCView.x86.exe for Windows OS) was located which shows various bit-level mappings for the webcam and can be instrumental in reverse-engineering the Logitech's interface. Table 5-2 shows these irregularities in detail. Fortunately, all parameters were successfully adjusted in this research project, except Exposure which needs to be adjusted manually in some cases. The following describes the best settings obtained for the prototype system – all experimentally determined.

Table 5-2: Irregularities in step size in the Logitech C920 camera

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>INPUT VALUES RECOGNIZED THROUGH THE OPENCV INTERFACE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exposure</td>
<td>MIN -1 -3 -3 -4 -4 -4 -5 -5 -5 -6 -6 -7 MAX</td>
<td>Irregular step. Some settings unavailable</td>
</tr>
<tr>
<td>Gain</td>
<td>MAX 255 250 245 240 235 230 227 221 216 ...6 3 0 MIN</td>
<td>Variations in step.</td>
</tr>
<tr>
<td>Zoom</td>
<td>MAX 180, 180, 180, 180, 180, 180, 180, 180, 180, 180, 180, 180, 180, 180, 160, 140, 120, 100 MIN</td>
<td>Irregular step. Some settings unavailable.</td>
</tr>
</tbody>
</table>

Exposure is set to 3 steps from the MAX value (see the range in Figure 5-2 where three steps vary the settings from -5 to -7) and depending on the light source intensity, the exposure can be lowered by a value of 2 or increased by a value of 1. Further increments in the exposure setting cause a significant drop in the camera's frame rate (10 fps) – as expected for low shutter
speeds – and renders the input unusable for tracking purposes. The correct Exposure setting has to be confirmed empirically for different camera models, although there appears to be a strong correlation for cameras from the same manufacturer.

Gain is set to the value 223. The MAX value can also be used; however, decreasing from that value by up to 10% expands the black colored areas. Decreasing beyond 10% causes the surrounding noise to resurface.

Brightness is always set to the MIN value. Increasing it up to 30% does not affect detection. Beyond 30% image features start to fade out.

Contrast is set to 31. The MIN value can also be used, but a slight increment from the MIN value lowers the noise.

Color Intensity (a.k.a. Saturation) is set to 215. Setting it to MAX promotes good pupil detection, but introduces significant noise for eye and nose detection, so a compromise needs to be made.

White Balance is set to 3949. This value creates the best contrast between the pupils and the rest of the image. Using an incandescent light source (2000 K) does not affect the performance.

Zoom is set to 180 so that the height of the face covers the screen from top to bottom. The camera's zoom function up-scales the image better than comparable software solutions and does not load the CPU.

Focus is set to the minimum value reflecting the macro camera mode since the camera is placed approximately 50 cm from the user. Variations of +/- 40% in distance do not significantly reduce the sharpness of the image.
It is also important to emphasize that setting individual parameters for this camera model needs to be carried out with 50 ms of delay, otherwise random errors or random camera settings occur. Also, when starting the camera, a safe delay of 500 ms is recommended for the same reason.

Although all of these parameters were set manually, an automated approach is easily devised through a set of interactive configuration steps:

1) For the factory camera settings, apply the Canny edge algorithm and compute the average width of edges for multiple frames as the user moves. The goal is to determine how large the motion blur is, which provides a measure of how useable the image is for a given Exposure setting. The Exposure setting is increased until either the motion blur is beyond the threshold or the level of details becomes too low, which determines the upper desired setting.

2) Set Gain to maximum.

3) Set Brightness to minimum.

4) Set Contrast to minimum.

5) Set Color Intensity (a.k.a. Saturation) to maximum.

6) Set the White Balance to the lighting source temperature.

7) Set the Zoom so that the height of the user's head fills the image vertically.

8) Set Focus to minimum.

These steps both ensure that enough of the features are detected and allow for further adjustment of only Gain, Contrast and Color Intensity within 20% of their settings, in order to achieve the best pupil, nose and eye segmentation. The weighting function in Equation 5-1 is constructed in
such a way that the Color Intensity parameter has the highest weight, followed by the Gain and the Contrast, as shown in the equation 5.1:

\[ W_f = \alpha \times \text{Color Intensity} + \beta \times \text{Gain} + \gamma \times \text{Contrast}, \quad (5.1) \]

where \( \alpha > \beta > \gamma \).

Figure 5-3 shows the original image with factory default camera parameters. Figure 5-4 shows the original image after the parameters have been adjusted for 3-DCS. Then the red channel information is extracted, filtering mostly the nostrils, pupils and eyebrows, shown in Figure 5-5. The eyebrows are visible in both the red and grayscale image; in this work the grayscale was used for detecting the eyebrow location. The 3-DCS mapping was successfully employed with two other cameras, the Microsoft LifeCam Studio and Logitech C920. Isolating the facial features using 3-DCS has no performance impact on CPU, which can be more effectively utilized to localize features.
Figure 5-4: Original image after adjusting the camera parameters for image segmentation.

Figure 5-5: The grayscale image of a red channel after the camera parameters are adjusted and red channel extracted from R'G'B'.
5.4 Region of interest and image preparation

For each individual feature a default region of interest (ROI) is set, which assumes that the user is located approximately at the optical center of the camera.

Alternatively, a well-known face detection algorithm can be employed for an initial detection of facial features.

Once the features are detected, they are tracked based on the last known position. In the case of pupil tracking, blink detection is implemented: if the detection of the pupil has been missed for 1 or 2 frames and then resumes, a blink is assumed. For all features, a 10 frame window is allowed in case the tracking is lost, and the algorithm tries to recover based on the previously valid position within the ROI. The ROI is always centered around the previous known point with sub-pixel precision. The ROI needs to be augmented in those cases when the center of the ROI is between pixels (see Figure 5-6a). The side effect of this augmentation is that the sliding window slides away from the center of the ROI. To compensate for this offset, the following steps have been implemented and are shown in Figure 5-6a through e. After the ROI is isolated, the resulting image is smoothed using the \textit{blur} function with a normalized matrix of size \texttt{ksize.width x ksize.height}. This matrix represents the kernel K which is shown in Equation 5.2.

![Figure 5-6a](image.png)

Figure 5-6a: Starting position for the ROI and the pixel. The pixel occupies half of two neighboring pixels.
Figure 5-6b: The pixel has been moved to the left by a half pixel. The ROI is moved accordingly so that the pixel is always positioned in the center.

Figure 5-6c: The ROI needs to be rounded for the next iteration. If the rounding causes the ROI to not be centered around the pixel, it is rounded up to the nearest whole pixel.

Figure 5-6d: The pixel moves by a half pixel to the right, once again occupying two neighboring pixels. The ROI follows.
The ksize.width is the number of columns and ksize.height number of rows in the kernel matrix. The kernel size is usually 3x3, 5x5 or 7x7; kernels larger than 7x7 have been found to slow down image processing on medium-resolution images (600x200 and above). The image is then upsampled using the OpenCV \texttt{resize} function employing the nearest neighbor interpolation which does not change the intensities of the neighboring pixels since high changes in intensities of pixels are desired. Also the nearest neighbor algorithm proves to be the fastest when compared to other OpenCV resizing algorithms with alternative interpolation methods. Good scaling factors are 3, 5 and 7 while higher numbers slow down the system. The ratio of scaling between the horizontal and the vertical axis is 1.

Then the image is converted from grayscale into a binary image where two levels – black and white – are assigned to pixels that are below or above the specified threshold, respectively. Using a global threshold value does not yield good results so the \texttt{adaptiveThreshold} algorithm is used. Adaptive thresholding calculates the threshold for small regions of different parts of the
image. The difference of these thresholds results from the varying illumination present across the image. The threshold value is calculated as the mean of the neighboring area. The size of a pixel neighboring area is correlated with the area of the detected feature. Another parameter in the adaptive threshold is the constant C which is subtracted from the mean of the neighboring area.

5.5 Pupil Detection

The pseudo-code in Figure 5-7 describes the detection and tracking of pupils. The width of the default ROI is set to 1/16 of the capture resolution width, and the ROI height is set to 1/20 the capture resolution height. In the case of 1280x720 resolution, this results in an 80x36 window. These ratios were found to correspond to facial proportions of pupils. Parameters in this algorithm that are adjustable in the GUI include:

- the choice of kernel from a selection of various kernels of different sizes and shapes including square, oval, ring and inverted oval
- adaptive threshold block size and constant C, used for precise adjustment of the conversion from a grayscale to a binary image
- morphology erode and dilate iterations used to lower the noise occurring around the edges of features
- morphology open and close iterations used to reconstruct the ellipticity of the features and to lower the noise

Figures 5-8a-c show an example of left pupil detection steps without image up-sampling; Figures 5-9a-c show comparable steps with up-sampling. As the pseudo-code shows, a
Set default ROI
Set global ROI to default ROI
Set default global pupil center
Set number of frames with undetected features
Set default kernel for morphological operations

For each frame:
  For each pupil:
    Isolate ROI
    Up-sample image (optional)
    Prepare image for thresholding
    Clean Stage 1 (optional):  
      morphological erode to clean the surrounding noise  
      morphological dilate to patch the inner holes
    Clean Stage 2 (optional):  
      morphological open to further lower the noise  
      morphological close to reshape the features circularity
    Find feature center:
      Find connected components
      The connected component with the largest area is the feature
      Find the centroid of the feature which becomes the new center
      Calculate the local offset from the previous center
      Global center is the local center downscaled by the up-scaling factor and with the added offset of the global ROI.
      If no features detected, increase number of frames with undetected features.
      If number of frames with undetected features becomes 2, blink detected; reset to 0.
      If the number of frames with undetected features becomes 10, tracking lost; restart at the beginning (Set default ROI).
      Update global ROI to reflect the offset between the current and previous local center
      Round the global ROI
    Find the center between detected pupil centers
    Compensate for the nonlinear center shift for different head angles

Figure 5-7: The pseudo-code for detection and tracking of pupils
combination of the morphological operators open, close, erode and dilate is used. Figure 5-8a shows a grayscale image of a left pupil after the 3-DSC mapping was performed and the red channel extracted from the BGR image. The constant C is adjusted until the round pupil is isolated from the background noise of the image, as shown in Figure 5-8b.

Figure 5-8a: Left pupil ROI without up-sampling

Figure 5-8b: Left pupil ROI after thresholding

Figure 5-8c: Left pupil ROI after applying morphological open operator with 5x5 disc kernel
In the next step a morphological open operator is applied to the image, where the kernel is a 5x5 disc shaped matrix. The resulting image is shown in Figure 5-8c. This step reconstructs the ellipticity (how circular the object is) of the pupil and stabilizes the center detection due to jitter caused by inherent noise in the image.

Figure 5-9a: Left pupil ROI with 3x up-sampling

Figure 5-9b: Left pupil ROI after thresholding

Figure 5-9c: Left pupil ROI after applying morphological erode and close operator with 5x5 disc kernel
5.6 Eye Detection

<table>
<thead>
<tr>
<th>Set default ROI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set global ROI to default</td>
</tr>
<tr>
<td>Set default global eye</td>
</tr>
<tr>
<td>center</td>
</tr>
<tr>
<td>Set number of frames</td>
</tr>
<tr>
<td>with undetected features</td>
</tr>
<tr>
<td>Set default kernel</td>
</tr>
<tr>
<td>for morphological</td>
</tr>
<tr>
<td>operations</td>
</tr>
</tbody>
</table>

For each frame:
  For each eye:
    Isolate ROI
    Up-sample image (optional)
    Prepare image for thresholding
    Find inner contours and fill as needed
    Clean Stage 1 (optional):
      morphological erode to clean the surrounding noise
      morphological dilate to patch the inner holes as needed
    Clean Stage 2 (optional):
      morphological open to further lower the noise
      morphological close to reshape the features circularity
    Find feature center:
      Find connected components
      The connected component with the largest area is the feature
      Find the centroid of the feature which becomes the new center
      Calculate the local offset from the previous center
      Global center is the local center downscaled by the up-scaling factor and
      with the added offset of the global ROI.
      If no features detected, increase number of frames with undetected features.
      If number of frames with undetected features becomes 2, blink detected;
      restart at the beginning (Set default ROI).
      If number of frames with undetected features becomes 10, tracking lost;
      restart at the beginning (Set default ROI).
      Update global ROI to reflect the offset between the current and previous
      local center
      Round the global ROI
    Find the center between detected eye centers
    Compensate for the nonlinear center shift for different head angles

Figure 5-10: The pseudo-code for detection and tracking of eye center
The pseudo-code for eye center detection (see Figure 5-10) is based on detecting the center of mass for eyebrows since they are the referent static point as the eyeballs move. It is somewhat similar to detecting pupil centers. The size of default ROI is set to the capture resolution width/8 and height/15 which in case of 1280x720 resolution results in a 160 x 48 window. These ratios were found to correspond to facial proportions of pupils. Due to the granularity of the eyebrow image, the eyebrow blob can contain openings within itself which need to be patched if a stable and precise centroid is to be calculated. Morphological operators are convolution-based.

Figure 5-11a: Left eyebrow ROI without up-sampling

Figure 5-11b: Left eyebrow ROI after thresholding

Figure 5-11c: Left eyebrow ROI after applying two morphological open operators with 5x5 disc kernel
operations and are fast if the image and kernel size are relatively small. In this instance, kernel sizes up to 7x7 do not negatively impact the overall performance.

The GUI-adjustable parameters are the same as in the pupil detection case. Figures 5-11a-c show an example of eyebrow detection steps.

5.7 **Head detection and pose estimation**

The head detection is based on detecting nostrils. The rotation angle of the head, which is defined as the head pose, is approximated as the distance of the nose (measured in pixels) from the center of the camera, multiplied by a factor $\alpha$. Recall that the user's head is assumed to be at the camera's optical center. The nose is a good feature to track for rotational angle of the head since it is the closest to the camera and travels the most during head movement. The pseudo-code is similar to the pupil and eye detection. Estimation of head pose is based on Equation 5-3:

$$\text{estHeadPose} = \alpha \times (\text{NoseCenter} - \text{OpticalCenter}),$$  \hspace{1cm} (5.3)

which works well for small head angles. Factor $\alpha$ is a function of the nose to optical center distance and the zoom parameter value, and it is determined experimentally. Figure 5-12a-c show an example of nose detection steps.

![Figure 5-12a: Nose ROI without up-sampling](image)
Kernel, kernel size, resize factor, binary threshold, and morphological operations are accessible through the GUI interface in the software for eye, nose and pupil image processing.

5.8 Eye gaze estimation

Eye gaze estimation is composed of head pose estimation, eye-brow center estimate and pupil center estimate as previously discussed. The horizontal distance measured in pixels from the pupil center to the eye-brow center, compensated for different head angles, provides the eye gaze offset information of the pupils. Figure 5.13 shows an example where the head is shifted to the right, eyes looking to the left. It is important to note that the camera is placed below the user's head, just below the chin. The system assumes that the user's head is aligned with the camera's optical center.
For the purposes of translation, only x-axis gaze estimation (horizontal scanning) is employed, while the y-axis gaze estimation is derived from y-axis head pose estimation. The target sentence to be translated is presented in a single-line format in this prototype system.

![Gaze estimation focal points](image)

**Figure 5-13: Gaze estimation focal points**

### 5.9 Translation module

The translation module has three main functions: translating a given sentence in real-time, detecting the user's cognitive load and presenting a possible solution. The solution can be in the form of an alternative translation for a word in question, or a synonym.

In the case of a reading assignment, the solution can also be presented as the word definition. The user interface in Figure 5-14 resembles a familiar translator's interface found in
Wordfast or Trados translation programs, separating the source and the target sentence. This spatial isolation has two purposes:

1) it is easier to measure gaze duration and to increase measurement validity;
2) the scenario reflects the initial human evaluation where individual sentences, instead of whole texts, are evaluated.

Figure 5-14: Translation interface in Wordfast, typical of an assistive translation system. Image acquired from [31] through Bing search engine under "Free to share and use" license.

In the Jezik system, the user types in a sentence or is provided with one in the source language editor (Figure 5-15). After the user presses Enter, the http get request is sent to Google
Translate, containing the source string, source language, user's unique pre-paid key and a desired target language. If the request succeeds, the server responds with a 200 OK status code and the JSON encapsulated object. The JSON (JavaScript Object Notation) object is parsed, the target string extracted and presented as the translated text. For each word in the target sentence the bounding rectangle is calculated, and the eye gaze tracker is activated.

![Visual display of the translation screen in the Jezik system](image)

Figure 5-15: Visual display of the translation screen in the *Jezik* system

As the user reads through the translated sentence the tracker measures how much time is spent on each word. The cognitive load detection algorithm is based on four parameters:

1. *First pass fixations*, which is the number of eye fixations for each word in the sentence for the first pass.

2. *Regressive fixations*, which is the number of eye fixations for each word in the sentence during regressions. For the purposes of this research, regression is defined
as re-reading the sentence or individual words after the user has read the last word on the first pass.

3. **Total fixations (adjusted)** per word.

4. Allotted reading time per sentence, which is a function of number of words and individual word lengths.

Total fixations (adjusted) are computed as shown in Equation 5-4, where the coefficients $\gamma$ and $\delta$

\[
\text{Total fixations (adjusted)} = \gamma \times \text{max(First pass fixations)} + \delta \times \text{max(Regressive fixations)}.
\]

are determined by experiment such that $\gamma < \delta$. It was found that $\gamma = 1$ and $\delta = 2$ are good values and can be further adjusted to reflect the user's reading speed. The function $\text{max()}$ returns the highest number of fixations per word in a sentence.

Figure 5-15 shows an example of a detected control word. The results of the first three parameters are shown under the STATS group box. In the First pass fixations row, the word "Otorhinolaryngology" had 12 eye fixations (6 more than the word "medical") and the highest number of fixations in the test sentence. In the Regressive fixations row, the word "Otorhinolaryngology" had 136 eye fixation (multiple times higher than the other words). The Total fixations (adjusted) row clearly delineates the word "Otorhinolaryngology" as the critical word. The red dots represent the eye gaze points during the allotted reading time per sentence. Only those dots that fall within the rectangular boundaries of words are considered to be fixations points.

The fixation duration varies as a function of lexical difficulty, as expected [32]. If the Total fixations (adjusted) has the largest value in the sentence, a control word has been detected. The
solution is presented above the word in question, in real-time, by accessing the external, hand-developed dictionary. The number of fixations can differ somewhat from the true number of fixations due to the camera's frame rate variance.

Other options on the translation screen interface include:

- Selector for different language pairs.
- Selector for hiding / showing gaze points during reading.
- An indicator on the availability of the external dictionary status (Google Translator).
- Processing speed in fps.
- Selector for reading / translation mode. In reading mode, the target language editor is collapsed.

Nine Croatian test sentences have been composed for the reading test, each containing one low-frequency (LF) word or a highly unanticipated word, and high-frequency (HF) words. Some of the chosen words are not part of the Croatian corpus (i.e. not part of the current Croatian language) but are pending review by Croatian linguists for inclusion in the language. As such they should be unknown to the user. A similar situation occurs for the word "google".
6 RESULTS

6.1 Camera resolution dependency

The eye gaze algorithm has been tested on different camera resolutions: 640x480, 800x600, 1024x768, 1280x720, 1600x1200 and 1920x1080. It seems that there is an inherent jitter approximately 1-2 pixels in radius at 1280x720 when localizing centers of individual features. In higher resolutions the jitter is less visible which in effect increases the precision of the feature centers.

The results with two other tested cameras, the Logitech C270 and Microsoft LifeCam Studio, were almost as good as the results of the Logitech C920 camera.

6.2 Pupil detection precision

Ground truth for pupil center detection was performed manually by zooming in on the image and manually locating the pupil center. The pupil location detected by the algorithm was compared to the manually detected pupil location and the difference in pixels was considered jitter. The jitter was 1-2 pixels in radius occurring 3-5 times per second.

The detection performed well even for extreme angles when the user moves his eyes all the way to the right or left. There are some losses of tracking if the user looks up above the screen, which is related to the positioning of the camera, not the detection algorithm itself. Also, the
algorithm uses a linear model for eye movement, since the angles presented in the reading tests were small enough to be approximated with a linear model.

6.3 Head pose and eyes detection precision

The head pose distance was measured manually by zooming into the image and manually determining the number of pixels between the camera center and the nose center. The difference between the algorithm head pose measure and the manual head pose measure was defined as the head pose jitter. The jitter was 1 pixel; it would occur 1-2 times per second and was less than in the case of pupil detection. It was observed that when the user rests his head on a solid object, the jitter disappears. The head pose estimation allowed reliable detection of the word under gaze, except for the extreme angles where the user looks off the translation screen. The reason for that is the increased deformity of nostrils at larger left-right head angles and the linear estimation of the head movement.

6.4 Eye gaze precision and speed

Any estimation of eye-gaze offset (see section 5.8) necessarily includes the jitter present in the eye-brow center measurement, the pupil center measurement and the nose center measurement. Another crucial factor which affects the jitter is the variation in location of the centers of the eyes, pupils and nose for different head poses. The result is eye-gaze jitter of approximately 30 pixels determined by comparing the manually detected location with the algorithm output, which for 1280 pixels of horizontal resolution translates to 2.34% of uncertainty in pixel coordinates. To put it into perspective, on a 1280 x 720 test screen, a 28 point font is best for a useful detection to the scale of an individual word. Of course, precision can be enhanced with more complex estimation algorithms and head pose models or template
matching techniques, but speed is a crucial factor in detecting eye movements, and this system has the necessary speed.

Recall that the primary emphasis of this thesis was to determine if the Jezik system was able to track eye gaze, detect translation inaccuracies and perform in real time on a standard personal computer. Indeed, these results were achieved as demonstrated in the following column. The tracking algorithm was run on a 6-year old Intel i5 560Mobile CPU paired with 533 Mhz DDR3 RAM employing a single core only. The execution time results are shown in Table 6-1. These results were obtained by measuring CPU tick count immediately before entering an

Table 6-1: Individual algorithm processing time

<table>
<thead>
<tr>
<th>ALGORITHM</th>
<th>MIN TIME (ms)</th>
<th>MAX TIME (ms)</th>
<th>AVG TIME (ms)</th>
<th>CAPTURE RESOLUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nose</td>
<td>0.17</td>
<td>0.44</td>
<td>0.21</td>
<td>640 x 480</td>
</tr>
<tr>
<td>Eyes</td>
<td>0.27</td>
<td>0.39</td>
<td>0.30</td>
<td></td>
</tr>
<tr>
<td>Pupils</td>
<td>0.14</td>
<td>0.27</td>
<td>0.16</td>
<td></td>
</tr>
<tr>
<td>Eye gaze estimation over a 5 second observation (150 iterations)</td>
<td>0.66</td>
<td>1.23</td>
<td>0.74</td>
<td></td>
</tr>
<tr>
<td>Nose</td>
<td>0.34</td>
<td>0.57</td>
<td>0.39</td>
<td>1280 x 720</td>
</tr>
<tr>
<td>Eyes</td>
<td>0.49</td>
<td>0.69</td>
<td>0.54</td>
<td></td>
</tr>
<tr>
<td>Pupils</td>
<td>0.23</td>
<td>0.52</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>Eye gaze estimation over a 5 second observation (150 iterations)</td>
<td>1.23</td>
<td>1.72</td>
<td>1.36</td>
<td></td>
</tr>
<tr>
<td>Nose</td>
<td>0.56</td>
<td>0.73</td>
<td>0.58</td>
<td>1920 x 1080</td>
</tr>
<tr>
<td>Eyes</td>
<td>0.67</td>
<td>0.85</td>
<td>0.67</td>
<td></td>
</tr>
<tr>
<td>Pupils</td>
<td>0.29</td>
<td>0.71</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>Eye gaze estimation over a 5 second observation (150 iterations)</td>
<td>1.83</td>
<td>2.64</td>
<td>1.95</td>
<td></td>
</tr>
</tbody>
</table>
algorithm loop and again after the loop has ended, subtracting the tick count and dividing by
CPU tick frequency. The corresponding OpenCV functions are \texttt{getTickCount()} and
\texttt{getTickFrequency()}. There is a linear correlation between the resolution, ROI and detection
speed. An increase in resolution will proportionally increase the average tracking speed. For a
160x72 nose ROI, detection time varies between 0.34 ms and 0.57 ms. Cutting the nose ROI by
half decreased the detection time to 0.15 ms and 0.35 ms. Similar results were obtained for eye
and pupil detection as seen in Table 1. Overall eye gaze estimation took between 1.23 ms and
1.72 ms per frame. If the longest time is taken into account, this gives the ability to track
\[
\frac{1000 \text{ ms}}{1.72 \text{ ms}} \times 1 \text{ fps} = 581 \text{ fps},
\]
which would satisfy the precise tracking of saccade durations which range as low as 20ms [30].
This can be further increased if each of the feature-tracking algorithms were assigned to a
separate CPU core, since feature detectors are independent of each other making them good
candidates for parallelization on a multi-core system. Also the average time is closer to the min
value rather than the max value, suggesting that there are occasional interruptions in the
continuity of the program execution, which might be caused by a 100ms GUI update period.

The Logitech C920 provides 30fps by default for most resolutions, although the frame-rate
fluctuated even when there were no other devices connected to the USB controller. For some
lower resolutions it is possible to force the camera's fps parameter through the OpenCV interface
to 60fps. However, the question remains whether the camera's CMOS is fast enough to supply
fresh data every \(~16\text{ms}\) or if the camera's interface is just providing duplicate frames.

It is also important to mention that there is no pattern matching or training data required
for the user to start using the system, it just works. In addition, there is sufficient unused capacity
to allow the inclusion of additional features in the *Jezik* system as will be described in the future research section.

### 6.5 Comparison with commercially available tracking systems

This system run times as just outlined can process ten times the amount of data used in these tests and still run in real-time, giving rise to the possibility of using a higher-speed camera for more precise saccadic eye tracking. While high-end commercial tracking solutions offer speed and relatively user comfort, they employ infrared illumination which is unacceptable for the everyday translation job. In this research, the existing restriction on the user's translation movement from the optical center of the camera can be enhanced by incorporating the geometric model for head pose estimation. The geometrical model for head pose estimation would require the use of the mouth tracking as explained earlier.

### 6.6 Different lighting conditions

For any given test of the *Jezik* system, while maintaining acceptable performance, increasing the light intensity makes it possible to decrease the exposure time, which in return improves the response time (less motion blur effect) of the camera thus effectively increasing the frame rate. The correlation between light intensity and exposure time is based on equation 6.1:

\[
\text{camera response time} = \text{light intensity} \times \text{camera exposure time}. \tag{6.1}
\]

On the other hand, we can lower the light intensity to a certain degree after which the frame rate is degraded to a couple of frames per second, and the motion blur effect becomes significantly pronounced. The system is relatively independent of the light source – it will operate well under fluorescent, incandescent or direct sunlight, with light intensity guidelines
described in section 4.1. It seems that there is a dependency on the angle of illumination incidence. For the best results, ring-light illumination (arrangement of an LED array in a ring) and in-line with the optics should be used since this configuration reduces shadows and offers relatively even illumination [14]. Other good illumination includes directional and diffuse types which reduce the glare. Directional illumination becomes uncomfortable to the user during long-time exposures. The system also performs well under direct sunlight, but only if the sunlight is directed frontally at the user and not at the camera sensor.

6.7 Accuracy of the cognitive load measurement

Figure 6-1 shows the Jezik system translation in use. The two images above the translation screen show the tracking of user's facial features and gaze estimation in real time. Recall that the camera is placed below the user's head, and although it appears as if the user is looking up, the white dot in the lower left quadrant shows the head pose, and the dot next to it the eye gaze. The translation example shows the accumulated gaze points as red dots. The first word "Otorhinolaringology" was a control word and it was clearly detected as such as noted by the highest number of Total fixations (adjusted). The solution was presented in the form of definition, above the word "Otorhinolaringology".

The magnitude of the Total fixations (adjusted) is higher for LF words (ones occurring 4 times per million) and for unexpected words (out of context words). Re-reading the sentence is also indicative of the overall processing difficulty. Table 6-2 shows the detection for various positions of the LF control words and unexpected words within each of the test sentences. Some inaccuracy of the Total fixations (adjusted) is attributed to the camera's variable frame-rate causing the accumulation of a number of fixations on words for which the frame-rate was higher
at the moment of reading. Other inaccuracy is related to the head pose estimate which works best when the user's gaze is at the same level as the sentence to be read. This orientation results in primarily horizontal motion of the head eyes. This restriction can be reduced by using a geometrical head pose model, as indicated earlier, in future work. The system performed as expected irrespective of the location of the control words in the sentence. Control words in test sentences 8 and 9 are highly specialized, rarely used words of unusual length, and thus have a higher number of fixations from the other control words, as one would expect.

The overall subjective performance of the system – except for limited body movement – in presenting the translations and definitions of the control words was satisfying. The two wrongly detected words would appear to be due to the following items:

Table 6-2: Total fixations (adjusted) for various LF word scenarios

<table>
<thead>
<tr>
<th>TEST SENTENCE No.</th>
<th>CONTROL WORD POSITION WITHIN A SENTENCE</th>
<th>TOTAL FIXATIONS (ADJUSTED)</th>
<th>CONTROL WORD DETECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Middle</td>
<td>8</td>
<td>Detected wrong word</td>
</tr>
<tr>
<td>2</td>
<td>End</td>
<td>42</td>
<td>Success</td>
</tr>
<tr>
<td>3</td>
<td>End</td>
<td>74</td>
<td>Success</td>
</tr>
<tr>
<td>4</td>
<td>Middle</td>
<td>66</td>
<td>Success</td>
</tr>
<tr>
<td>5</td>
<td>End</td>
<td>46</td>
<td>Success</td>
</tr>
<tr>
<td>6</td>
<td>Beginning</td>
<td>58</td>
<td>Success</td>
</tr>
<tr>
<td>7</td>
<td>Middle</td>
<td>8</td>
<td>Detected wrong word</td>
</tr>
<tr>
<td>8</td>
<td>Middle</td>
<td>139</td>
<td>Success</td>
</tr>
<tr>
<td>9</td>
<td>Beginning</td>
<td>116</td>
<td>Success</td>
</tr>
</tbody>
</table>
- the use of the limited number of sentences and the user's eventual familiarity of the sentences, thereby reducing the fixation time
- variation in frame rate of the camera which can be overcome by using the higher frame rate camera
- automatic calibration due to imprecise calibration which could be corrected by a geometric model for automatic calibration.

Figure 6-1: The Jezik system in use
7 CONCLUSION AND FUTURE WORK

7.1 What was done

A cognitive translation system employing a single, visible spectrum tracking detector has been built for assistive translation and reading purposes. No additional external equipment was used except for an inexpensive, single web camera. The system's interface adheres to the OpenCV standard interface and it is suitable as a platform for further academic research.

The Jezik system performed well for the modules designed and tested. The exception was detection of dark skin users and users with dark brown eyes, in which case a different color mapping is required to maintain the Jezik system performance. The pupil will still be present in the red channel of the adjusted system, but time did not permit extensive testing for this case.

7.2 Future research

The foundation for a cognitive translation assistant has been constructed, and it can be extended through further development. It would be interesting to investigate how the system would perform on a diverse group of users with different backgrounds and native languages. The following sections outline some of the improvements that could be considered.
7.2.1 Improving gaze detection

Obtaining pixel-perfect feature detection is crucial for eye gaze precision. The current system’s speed and small ROI leave plenty of room for additional implementations of feature detection algorithms such as pattern matching or supervised training in order to stabilize feature detection precision.

Some of the OpenCV functions are GPU ready, and individual feature algorithms can be parallelized. Based on the computation performed, it is estimated that the speed up factor would be close to the number of parallel executions. Using an elliptical ROI (versus rectangular) for the features can further reduce the processing time.

Using the distances and angles between the parts of facial features (Figure 7-1), the current head pose model can be upgraded to a geometric head pose estimation model which provides the

Figure 7-1: Final results showing the basis for the geometric head-pose model
detection of the roll, pitch and yaw rotation (Figure 7-2). This model would allow the user to exert three degrees-of-freedom head movement.

Murphy-Chutorian and Trivedi [33] describe the geometric head-pose model and review other existing models for head-pose estimation.

The quality of the tracker can be improved through feedback and the automatic adjustment of parameters, as explained in this research, to allow for dynamic changes in lighting conditions.

Figure 7-2: Outline of the calculation steps to determine the head pose detection (Figure extracted from [33])

Increasing the overall gaze precision opens the possibility for more flexibility such as lowering the font size and reading of multiple lines.

In imaging components, optical aberration due to varying light sources such as incandescent, fluorescent, or direct sunlight causes lateral color shift and chromatic focal shift, thereby causing problems when trying to resolve details and gain information in imaging systems. To avoid this issue, monochromatic illuminations of a single wavelength, such as an LED light, can be used to mitigate or eliminate the issue and lower the noise levels.
The sharpness parameter is not provided with the Logitech C920 interface, but can be controlled through OpenCV. Sharpness defines edge contrasts and should be explored if it can offer any benefits.

Information has been made available on how to access a similar Logitech webcam in Bayer-mode, i.e. to gain access to individual sub-pixel elements and the raw sensor data [34]. This would open a myriad of modes and settings to explore.

Future research may focus upon investigating if the web camera's market offers a medium to high-resolution 60fps device. Saccades have a duration as short as 20ms (and even a 10ms micro-saccade subcomponent) [35].

7.2.2 Extending the translation module

Cognitive load detection is based on the reading process, and it does not matter if the user is a translator or a reader. It should therefore be possible to explore the use of the Jezik system in assisting beginning or developing readers. The goal would be to show that such an assisted reading system can accelerate the development of reading abilities.

The logic in the translation module for a translation task or a reading task is the same and it can be extended onto other applications that utilize the concept of eye tracking and reading. These include:

- Reading assistants for readers with certain disabilities, such as lexical processing disability, or slow readers.
- Reading assistants for children who are learning to read; an example sentence is "Mom and dad are happy" where the word "Mom", "dad" or "happy" can be
replaced with a representative picture of a mom, dad or happiness, in real-time, triggered by detection of cognitive load caused by an individual word.

- Measuring the user's reaction to the text metrics such as size and font, and adjusting them dynamically to ones the user is more comfortable with.

Incorporating additional eye movement parameters, such as when the user looks off the screen, or recognizes complex patterns within the paragraph, could further increase the robustness of detecting the cognitive load.
8 BIBLIOGRAPHY


http://www.photomet.com/library_enc_gain.shtml


[23] Tom Heyman, Vincent Spruyt, and Alessandro Ledda, "3D Face Tracking and Gaze Estimation Using a Monocular Camera".


APPENDIX A. GLOSSARY OF TERMS

cornea – Transparent membrane covering front of eye: the transparent convex membrane that covers the pupil and iris of the eye.

fps – Frames per second.

GUI – Graphical user interface.

high-frequency (HF) words – High-frequency words occur 160 times per million [7].

lexical meaning – Dictionary meaning of word: the meaning of the base word in the set of inflected forms paradigm.

lexical processing – Cognitive process of a word recognition.

low-frequency (LF) words – Low-frequency words occur 4 times per million [7].

regression – A saccade that moves the eyes backward in the text, to read the material previously encountered.

ROI – region of interest; a subset of pixels of an image.

saccade – A rapid irregular movement of the eye as it changes focus moving from one point to another, e.g. while reading; movements between eye fixations.

source language – refers to the original language of a text.
target language – refers to the language the text is translated into.

total fixations – the sum of all fixations, including regression, on a word [6].

translation memory – a database that stores segments of text (a sentence, a paragraph) that have previously been translated and organized in language pair called translation unit.
APPENDIX B. TEST SENTENCES

CROATIAN (Control words are italicized)

1. Najrasprostranjeniji suosnik ima otpor 750 ohma.

2. Ona se vratila iz zapozorja.

3. Obuci svoje cipele dostojanstveno i bezoklijevno.

4. U ovoj rečenici nejasnica izaziva ponovno čitanje.

5. Po uputama, čovjek je predao affidavit.

6. Ablacija potkoljenice je bezbolan postupak.

7. Očigledno je koliko je naš sustav aberirao.

8. Isčašenje abartikulacije je učestalo kod osoba s artritisom.


ENGLISH TRANSLATION

1. The most widespread coaxial cable has the resistance of 75 ohms.

2. She returned from backstage.

3. Put your shoes on with dignity and without hesitation.
4. In this sentence the unknown word causes re-reading.

5. Following the instructions, the man surrendered the affidavit.

6. Ablation of the lower leg is a painless procedure.

7. It is obvious how our system has aberrated.

8. Ankle dislocation is frequent in people with arthritis.

9. Otorhinolaryngology solves a large number of medical cases.