Spring 2017

Assistive Technology for Hard-of-Seeing Guitarists

Alejandro Ramirez

ASSISTIVE TECHNOLOGY FOR HARD-OF-SEEING GUITARISTS

A thesis submitted to
Regis College
The Honors Program
in partial fulfillment of the requirements
for Graduation with Honors
by

Alejandro Ramirez

May 2017

Thesis written by

Alejandro Ramirez

Approved by

Thesis Advisor

Thesis Reader or Co-Advisor

Accepted by

Director, University Honors Program

Table of Contents

Literature Review

1) Arky

Arky writes an informative webpage exposing the umbrella of unique visual processing challenges. As a result of the vast differences among visual issues, assistive technology is usually as specific as the visual issues it was created to help.

2) Chityala

Image processing and acquisition using Python details usage of image processing to manipulate images and discover information from these changes. Chityala's image processing concepts work with the Python programming language. Chityala's Python scripts inspired many concepts found in this paper.

3) Dreon

Middle school teacher Dreon uses digital storytelling (a series of YouTube videos) to enhance classroom material. Proponents argue students are attracted to digital storytelling for the ability to access teaching material at home. However, opponents question student's equal access to digital storytelling, arguing it enhances learning only for students whose families can afford computers. Similarly, teaching material on the interment marginalizes some aspiring guitarists, especially visually impaired guitarists.

4) Godsey

Godsey argues too much technology in the class room changes a teacher's role into a facilitator. These changes expand beyond the classroom in the form of new markets for teachers to purchase lesson plans. Some teacher's feel their lessons are reproducible and replaceable. Godsey's current information between the relationship of technology and education brings into conversation the impact of assistive technology upon guitar educators.

5) Jones

The ever-changing effects of globalization on the music industry (Jones, 2007, p. 1) forces music teachers to continuously adapt their teachings. Nowadays students prepare by learning digital formats of music and social media self-promotion. Musicians need to use the internet as a result of the digital age. Jones argues musicians must serve the modern needs of society. Jones's argument brings in light the modern needs of guitarists.

6) Knight

A Framework for Recognizing Hand Gestures suggests "the use of hand gestures in free space is often seen as an intuitive next step in the progression of user input technologies," (Knight, 2010, p. 1). In other words, communicating with machines no longer relies on an object, surface, or button. Everyone will benefit from this software through faster and easier communication with computers. For this paper, the focus of computer recognition software centers on assisting people who cannot use traditional user input methods.

7) The National Institute on Deafness and Other Communication Disorders

NIDCD's online article details assistive devices and their impacts on those with communication issues. Many assistive technologies help improve digital communication, phone communication, and face-to-face communication.

8) Nielsen

The Impact of the Digital Age on Professional Musicians covers the impact of each technological step in the music industry up to the current decade (phonographs, radios, television, vinyl records, cassette tapes, compact discs, and digital music). Nielsen argues the musician deals with positive and negative repercussions of the digital age. Positive outcomes stem from more control on the creativity of music. Negative outcomes stem from illegal downloads. Nielsen's paper captures the complexity in the music industry when new technology approaches music. Thus, assistive technology for guitarists can produce positive or negative repercussions.

9) Pramada

Pramada demonstrates inclusion for the visually-challenged community by exploring how computers can "understand human language and develop a user friendly human computer interface (HCI)," (Pramada, 2013, p. 45). Some of their sections (Camera Orientation, Camera Specifications, RGB Color Recognition, Color image to Binary image conversion, Thresholding, Coordinate Mapping, and Pattern Matching Algorithm) inspired the algorithm used in this recognition software. Inspiration from these sources impacted this project's camera type, camera positioning, lighting, sample size, thresholding, and hand location in pictures.

10) Pring

Pring's study surveys thirty-two families with and without a child who has septo-optic dysplasia to study the stigma of septo-optic dysplasia upon music interest. Pring's findings suggest children with optic impairment have significantly greater interest in music. Thus, assistive technology these children the ability to paly music.

11) Qing

Teachers and students are most impacted by technology in the classroom and Qing studies the differences in their opinions. Overall, Qing finds negative perspectives from teachers and positive perspectives from students. A phenomenon called "Oversold and Underused" is growing due to school districts investing more money in technology but teachers are refusing to incorporate a high amount of the technology in their lessons.

12) RNIB

RNIB offers curriculum guidelines for music teachers with blind and partially sighted students. Their documentation addresses music notation, teaching strategies, and further resources.

13) Rokade

Rokade aims to "enable signers to interact with non-signers without requiring an interpreter. Such a system is useful when regular speech and audio are infeasible or limited, such as scuba diving, floor trading, paramilitary engagements and so on," (Rokade, 2016, p. 381). Rokade connect signers and non-signers by constantly detecting hand gestures through video input. This paper builds upon Rokade's efforts to accurately detect the rotation of the user's hand. Accurately detecting the rotation of a person's hands is essential for assistive technology to guitarists learn correct form.

14) Stafford

Stafford argues the evolution of music changes how musicians distribute music and how consumers legally or illegally attain music. Even social media has developed into a platform for musicians to easily share their music resulting in more control over their product. Since changes of technology impact the musician and the consumer this paper explores how assistive technology can give control to visually-impaired guitarists.

15) Willings

Willing argues that the majority of assistive technology used in schools and homes require connection to a computer. Thus, assistive technology does not lend itself to be transportable. This paper explores how mobile assistive technology can offer more benefits to its consumers.

16) Yuan

Yuan’s team helps blind folks feel included within the realm of video games by creating a version of Guitar Hero tuned to their needs. Their efforts inspire this paper to create a product that helps blind guitarists feel included with other guitarists, musicians, or students in their music class.

Preface: Why Music?

I believe, we remain in the presence of one of the last surviving universal truths. Music is an energy every human can participate with, create with, change with, and get lost within. Music transcends languages and adds dimensions to words and sounds. Music is an exquisite energy impacting our feelings, thoughts, and actions (RNIB, 2013).

As we find this energy, we tend to selfishly ask; “What can this energy do for me? How can I use it to my advantage?” We ask the wrong questions. One should not delegate instructions or orders to music, but rather converse with music. In our century, abstract ideas and experiences are pushed toward the margins of importance. We are largely concerned with music for reducing stress, anxiety, and blood pressure or perhaps we focus on the social and health benefits. We expect and exclaim “results,” but music is partly tangible and partly ethereal. There exists an undeniable possibility for music to shape our identity (RNIB, 2013) given enough searching. We must be comfortable with the part of our identity sculpted by music, especially as we collect experiences. With each new experience the same piece of music can offer a new journey. Each new journey adds to our identity and reveals part of our identity to ourselves. So why music? *Music adds to our identity, complicates our identity, and reveals parts of our identity to ourselves and others.*

Introduction

We need to deeply understand the needs of hard-of-seeing guitarists before creating any product to assist their needs. Hopefully through increased awareness more efforts can improve the quality of assistive technology. This paper dives into the multiple layers (music industry, technology, and assistive technology) affecting aspiring guitarist who experience visual challenges. These layers are interrelated beginning with the cycle between the music industry and technology. The music industry defines the standards for guitarists and new technology sets new trends which in turn impacts the music industry. As a result of the cyclical relationship between the music industry and technology, music education continuously changes because music education needs to prepare students for the music industry. Thus, the guitarist needs adapts to the changes in the music industry.

This paper responds to the lack of growth for assistive technology. We can increase inclusion for visually-impaired guitarists by providing more assistive technology. In response to the need for more technology this paper explores how image processing plays a role in software recognition for guitar chords. Creating assistive technology which recognizes guitar chords can help aspiring guitarists with visual challenges adapt to the changes in the music industry and music education.

Chapter 1

Music industry in response to technology

History shows an evolution of technology triggers an evolution in the music industry. The rapid and unescapable shift to digital music, starting in the mid 1990's, changed consumer's way of acquiring music and the ways in which musician's create music (Stafford, 2010, p. 112). As media evolved into digital files, the era of listening on computers and MP3 players dominated over live music, vinyl records, cassettes, and CD players (Stafford, 2010, p. 113). In summary, the way people interact with music changes as technology changes music (Stafford, 2010, p. 115).

Musicians no longer require vast amounts of resources to create a project. Most musicians feel "the digital age has brought more connection, less artistic compromising, more musical freedom, more musical learning opportunities, and more access to new musical ideas" (Nielsen, 2015, p. 17). The explosive nature of technology allows more musicians to pursue a passion for music. Evidently, approximately 75,000 albums are created in a year (Nielsen, 2015, p. 3). Technology is quickly accommodating to the needs of all musicians. Embracing new technology benefits everyone looking to create and enjoy music (Stafford, 2010. p. 118).

As we look for "new and improved ways to do even the simplest things" (Stafford, 2010, p. 118) we need to study how these changes impact those in more need of technology. Before discussing how the digital era can help the visually-impaired guitarist we must discuss where music education stands in light of technological changes.

Digital age and music education

The music industry is incredibly competitive, notably after the boom in technology. More than ever, aspiring musicians needs to learn valuable skills "to enter, survive, and thrive in the creative economy." (Jones, 2007, p. 12). Educators must evolve their teaching strategies to respond to the growing creative economy because successful musicians relate to their community (Jones, 2007, p. 14, 18). For example, music fans access music digitally more than any other medium. Therefore, music education needs to teach students how to use digital assets to create and promote their own voice.

An example of digitalizing education and music is YouTube videos. Specifically for guitar, several entrepreneurs capitalized on this market by creating a type of curriculum through tutorial videos usually inspired by popular songs (Dreon, 2011). Digital resources help guitarists learn quicker. The use of digital resources saw enough growth for Millersville University to teach classes on "how technology can be integrated in different content areas using sound pedagogical approaches" (Dreon, 2011). From independent bloggers developing their own business to formal college programs, the boom in technology is changing the learning of music and increasing access to learning material (Dreon, 2011).

Chapter 2

Accessibility issues with technology

Difficulties with vision origniate from either the brain or the eyes. The brain processes information from the eyes meaning weaknesses within the brain can lead to visual processing disorders (Arky, 2014). Whereas problems in the eyes interfere with receiving information and passing it to the brain. There are at least eight types of visual processing issues and people are not limited to one type (Arky, 2014). Furthermore, people struggle to find symptoms because vision tests might not detect processing issues (Arky, 2014). Thus, increasing awareness leads to more support and technology for vision issues. Vision issues affect learning, socializing, and coordination skills (Arky, 2014) impacting several layers of a person's life, possibly guiding feelings of loneliness and low self-esteem. Hopefully assistive technology helps those with vision difficulties conquer these obstacles and emotions.

An assistive device or adaptive device is a "device or service that increases participation, achievement or independence" to an activity (Arky, 2014) or in other words "refers to any device that helps a person with hearing loss or a voice, speech, or language disorder to communicate" (NIDCD, 2016). In summary the technology considered to be assistive meets a person's "unique learning needs" (Willings, 2016). For example, *Tech Finder* prompts a user to input their age, visual issues, and technology of interest to help find tailored technology (Arky, 2014). Current technology includes, but is not limited to, keyboarding instruction, video magnifiers, apps, lightboxes, and braillewriters (Willings, 2016), showcasing great options dedicated to meet a variety of needs. However, few assistive technologies exist related to music. Inventing and supplying this technology can encourage folks with vision challenges to “participate more fully in their daily lives” (NIDCD, 2016) through music.

Assistive Technology and Music

Music education needs to accommodate for hard-of-seeing musicians because musical instruments require “peculiar movement, dexterity and balance to operate them” (RNIB, 2013, p. 7). Aspiring musicians should find technology at schools or self-taught products to help them grow in skill. Musicians rely on building “skills and experience” in order to become more accurate and creative (RNIB, 2013, p. 6). Assistive technology can help build skill and experience. Learning music should be treated as any other subject, especially for younger musicians because a passion for creating music can be an important factor in developing a sense of identity (Jones, 2007, p. 5).There remains an unfulfilled need, an ignored need. “The question is whether or not we as a profession possess the commitment to live up to our responsibility” (Jones, 2007, p. 19) and serve our fellow musicians,

Music education performed within schools typically requires “Seeing the teacher... seeing other children in your performing group, seeing the instrument you are playing, and being aware of the reactions of those just listening are all part of music making in school” (RNIB, 2013, p. 6). Schools need to push for more assistive technology to provide equal opportunity (RNIB, 2013, p. 9), especially for students refusing to learn music at school due to lack of technology or feelings of exclusion. While music educators provide unmatched help, not all musicians can afford lessons or desire lessons. Teaching should be complemented by technology, but for aspiring musicians without teacher's instruction comes exclusively in the form of technology.

Displacement between digital age and hard-of-seeing guitarists

From consumption to creation, technology has evolved human interaction with music. Many guitarist benefited from this change, especially self-taught guitarists. However, guitarists with optic obstacles remain outside the benefits birthed by the 21st century. Although hard-of-seeing students typically have less music education access than fully-sighted students (Pring, 2005, p. 1), children with optic challenges show statistically significant more interest in learning music than "those who were fully sighted (Pring, 2005, p. 3). These children can use technology to enhance their experience with music shown by the video game Blind Hero (a modified version of Guitar Hero) developed by Yuan and company. Blind Hero shows evidence that translating visual stimuli for haptic (relating to the sense of touch) stimuli can produce similar gameplay experience (Yuan, 2008, p. 1). The developers replaced visual buttons on the screen with paper motors on a glove. The player would feel the motor move for a particular finger signaling to press down on the controller. Instead of looking at the screen to know when to press the button the motors in the glove cue the user's fingers. In this format players with optic challenges can also play the game. Replacing visual stimuli can "improve the quality of life of many disabled individuals who feel left out as the majority of the games are not accessible to them" (Yuan, 2008, p. 7). Hopefully more technology is created to include blind gamers and musicians.

Chapter 3

Role of assistive technology for hard-of-seeing guitarists

The role of assistive technology is for accommodation of needs rather than replacing music teachers. Technology's negative stigma in the classroom continues to rise as more teachers are converted to facilitators (Godsey, 2015) leading to underused technology (Qing, 2007, p. 377). Teachers should recognize students' view of technology as more of an enhancement for learning as opposed to a replacement for their teacher (Qing, 2007, p. 380). A great music teacher is irreplaceable but difficult to find. Assistive technology could be the medium in which visually-challenged guitarists learn the guitar (Qing, 2007, p. 387). Teacher support for assistive technology will expose more students to needed technology, which balances the music class. Preferably schools have enough staff to help students with vision challenges as much as fully-sighted students, but if not then teachers should encourage these students to seek assistive technology.

In conclusion assistive technology alone should be used alongside teachers, braille music, family, friends, and music by ear. Learning to play music for the optic challenged takes support, time, and technology. Although technology is an enhancement, support can be the turning point for many guitar students.

Ideal product

The ideal product allows for full participation in learning and playing guitar. The obstacles for learning guitar vary for each guitarist with vision issues. Some common guitar struggles are finger placement, strumming, reading music, holding the guitar, tuning, chord memorization, changing between chords, finger picking patterns, and most importantly practice. Ultimately if hard-of-seeing guitarists can practice regularly they can learn.

The ideal product would be affordable, transportable, and intuitive to use. These key requirements lead to accessibility. Affordability reaches as many people as possible which generates more awareness. The product should be small, easy to transport, and easy to use. Lastly the product needs an intuitive user interface following the example of phone applications accessible for hard-of-seeing customers. The user interface should incorporate voice recognition with voice commands to free up communication between the product and the consumer. Physically the product will look a bit bigger than a GoPro coming with an attachable tripod for different height settings to practice standing up. The front of the product will have a small built-in camera facing the user for recognizing guitar chords. The user will rely on the product for correction on finger position, chord positioning, and strumming. The product will provide sound recognition to listen to the chord or note being played and offer vocal response for correction. The product will also include a built-in tuner and metronome. Lastly, the product will be able to import songs and videos providing a play by ear approach. In summary the product should replicate a beginner guitar book for a hard-of-seeing guitarist.

Why the ideal product does not exist

Manifestation of the ideal product requires resources, experience, and time. Resources comes in the form of awareness to promote assistive technology in schools and money to invest in a product. Such a product requires teamwork among people with expertise in, video input, video recognition, programming, guitar playing, and other uncountable variables. Their teamwork will bring together many aspiring guitarist. Prototypes lead to the ideal product helping create the technology and spread even more awareness. As more awareness grows more investment is dedicated toward the product. These two concepts go hand in hand until a product comes into fruition. Each prototype should focus on one use-case such as a tuner, sound recognition, or chord recognition. Although each use case may take years, after enough refinement an ideal product should be available to the public.

Chapter 4

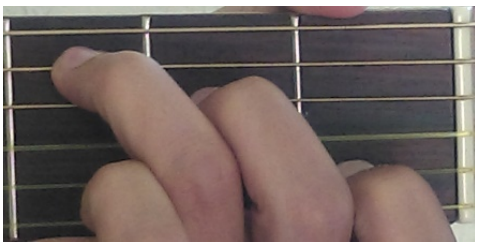
Contribution

The following section breaks down the software which recognizes a G-chord (in spirit of promoting awareness for assistive technology the Python code used in this paper is in the appendix). The code centers on ten pictures (test images) taken of a guitarist playing the G-chord. In half of these pictures the guitarist played the G-chord correctly and in the other half the guitarist playing the G-chord incorrectly. These test images are processed by comparing each against a control image of a correctly played G-chord. The software calculates whether each test image is a G-chord based upon the similarity between the test image and the control image.

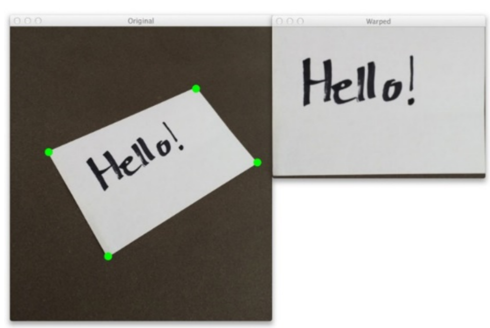
Firstly, test images are run through a program called Warp Perspective Transform (WPT) which prepares them for processing. In order to maximize precision when comparing images, the test images and control image should share similar characteristics. Imagine laying two images on top of one another. If these two images are the same size then it will be easier to compare them. Thus, we need to manipulate the color, size, and rotation of each test image to improve precision.

Preparation: Warp Perspective Transform

The images are gray scaled to minimize variance in color. Gray scaling means each image is only in shades of gray which is easier to compare than colored images. After gray scaling each image, the images are resized to 512 pixels by 242 pixels. Then, each image is rotated so the guitar neck is vertically aligned with the right edge of the image. The similar angle of the guitar neck between images results in similar finger placement within the images. Below is an example of gray scaling, resizing, and accounting for rotational difference (Before and after pictures are shown in the visual glossary on pages thirty-five to thirty-six).



Note, the program is not automated for the changes previously stated. The user executes WPT several times for a single image until the user is satisfied with the transformation. In other words, WPT is a program which helps the user manipulate the images through a trial-and-error approach. After each execution the original image and transformed image are displayed to examine the change as a feedback system. The user can run the program again with better coordinates after seeing the previous transformation. For real-time purposes Warp Perspective Transform should be automated. The algorithm should locate the corners of the neck of the guitar from the nut to the 3rd fret by itself. Since the sample size for this project is small the program was designed to be a skeletal setting, meaning the user has to change the values in Pts1 every time to test a new set of coordinates.



Before detailing how the rotational difference is calculated it is advised to view this example by Rosebrock. The green corners represent the original corners of the image which become the corners of the transformed picture. The image is rotated and cropped to minimize unwanted data. Calculating the rotational difference is done with a function called getPerspectiveTransform (GET). GET uses two tuples of coordinates. Each tuple contains four pairs and each pair represents an X-Y coordinate (top-left, top right, bottom-left, bottom-right. In other words, each tuple represents a rectangle and each coordinate pair within the tuple represents a corner of the rectangle. A picture of tuples are shown in the visual glossary under the name GetPerspecticeTransform. The two tuples are labeled Pts1 and Pts2. Pts1 represents four coordinates (corners) of the original image. Like the green corners in Rosebrock’s example. Pts2 represents what values the coordinates will become. Since all images are 512 pixels by 242 pixels Pts2 is the same for each transform. This means each image will be resized to 512 pixels by 242 pixels during the transformation. The GET function takes these two tuples and creates a 3x3 matrix to perform the transform from Pts1 to Pts2. The purpose of GET is to calculate the transformation matrix in order to correct rotational difference. The details of calculating the 3x3 matrix are outside the scope of this paper, however more information is located at the bibliography source titled Geometric Image Transformations.

The second function warpPerspective (Warp) takes three parameters; the image to be transformed a 2-D matrix, the matrix transform calculated from GET, and Pts2. Warp takes the transformational matrix applies it to the original image, resizing the original image to the width and height of Pts2. If the coordinates were chosen correctly the transformed image have gone through rotational and cropping measures. All images underwent this transformation in order to minimize unwanted data.

Global-Segmentation-Subtraction

Global-Segmentation-Subtraction begins with segmentation, then subtraction, and finally a quantization for the difference between a test image and the control image. Although the segmentation is executed prior to subtraction, the segmentation is used to enhance the subtracting. Thus, discussing subtraction first is vital before discussing segmentation. For each section subtraction and segmentation the different methods are described along with reasoning for each chosen method.

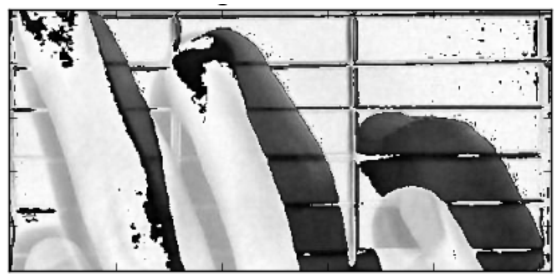
Subtraction

Images are stored as 2-D arrays (matrices) with hundreds of rows and columns (Chityala, 2014, pg. 68). Subtracting these vast matrices is a tedious calculation because subtraction between images requires pixel-by-pixel subtraction. Each pixel, in the 512 by 242 sized matrix, is compared to each respective pixel in the other image's matrix. The pixels contain values called pixel intensities ranging from zero to 255 (Chityala, 2014, pg. 137). Zero meaning white and 255 meaning black. The fingers are on the lighter side of the spectrum while the neck of the guitar is on the darker side. This paper explores Image module's three different methods for subtracting images; the Difference function, the minus operator, and the Subtraction function.

1. Difference Function

The Difference function "returns the absolute value of the pixel-by-pixel difference between the two images," (ImageChops). Since the difference is absolute the origination of the difference is disrupted. Was the original image darker than the test image in this pixel or was the test image darker? As a result the two images appear overlapped which is not the desired effect. The pixel intensities are difficult to differentiate between the original image and the test image. The difference function is undesirable because the function poorly reflects the difference in pixel intensities between the images.

2. Minus Operator



As seen through the following image this method effectively displays the subtraction. After subtracting the images this method causes all pixels near zero to change to 255. Pixels with similar intensity between the images are set to white. In other words, the difference between in the images is shown in white. The original image is shown in the background with the test image subtracted from it.

3. Subtraction Function

The subtraction method, "Subtracts two images, dividing the result by scale and adding the offset," (ImageChops). Similar pixels between the images will be darker, and contrasting pixels between the pictures will be lighter. The difference shows as darker intensity levels (opposite of the minus operator). The Subtraction method effectively differentiates the original fingers from the neck of the guitar, and shows the subtracted fingers as similar intensity as the neck. Thus, the algorithm only has to check the bright pixels from the original image. With the subtraction method, the algorithm can detect the difference by only focusing on the lighter pixels.

To improve the calculation's accuracy we need to further separate the lighter pixels from the dark pixels to better estimate the difference between two images. For this separation we turn to segmentation.

Segmentation

Segmentation helps decipher images by separating one image into multiple segments, particularly a region or regions of interest (Chityala, 2014, pg. 176). Separating the image yields better analysis and visualization of the area of interest. Usually these regions are called the foreground and background. For this project the fingers are considered the foreground and the guitar is the background. The three common methods for segmentation are Otsu, Adaptive Thresholding, and Watershed (Chityala, 2014). The following section details each method and reasoning for the chosen method.

Otsu

Otsu segmentation uses a threshold to create a black and white version of the image (for gray scale images pixel intensities range from zero to 255). A threshold is a value which each pixel intensity is compared to and is changed to white (255) if it is greater than the threshold, or turned to black (zero) if it is less than the threshold. For example, if the threshold is 128 then every pixel's intensity is compared to 128 and set to white if greater than 128 and set to black if less than the threshold.

The threshold is computed from the image's histogram (Chityala, 2014, pg. 178). The histogram of an image "is a graphical depiction of the distribution of pixel value in an image," (Chityala, 2014, p. 78). An example is available in the visual glossary. The histogram displays the distribution for pixel intensities. On the histogram, the pixel values start at zero at the origin and extend rightward to 255. A brighter image will have more distribution towards the right of the histogram and a darker image will have more distribution towards the left side of the image.

The optimal histogram for Otsu's method consists of two peaks and one valley like in the visual glossary (Chityala, 2014, Pg. 79). These two peaks typically represent the background and foreground. The valley represents the pixel intensities which separate the image. The lowest point of this valley is the pixel intensity which best separates the image. The lowest value of the valley is the threshold value. The threshold value maximizes variance between the background and foreground. Thus, Otsu's method works best for an image with a uniform background and uniform foreground.

Adaptive Thresholding

Adaptive Thresholding takes the math behind Otsu's and applies it to smaller sections within the image (Chityala, 2014, pg. 186). Instead of a global threshold, local thresholds are used for each section of the image. For each sub-image the local threshold is calculated through different methods such as mean, median, Gaussian, or other algorithms not discussed in this paper. For this project, the sub-image has a preset size to 40 by 40 pixels (if the size for the sub-image is increased to the length and width of an entire image then it is essentially an Otsu segmentation). The main usage of Adaptive Thresholding is that each sub-image uses a local threshold to better estimate that section of the image. As a result the Adaptive Thresholding performs better than Otsu's segmentation for images with greater variance in pixel intensities. A comparison between Otsu and Adaptive Thresholding is in the visual glossary. Adaptive Thresholding uses local thresholds in the example image which takes into account the different lighting for the input image resulting in a properly segmented image.

Watershed

Watershed segmentation uses region-segmentation over the entire image (Chityala, 2014, pg. 190). Watershed begins be separating the image into different regions. These regions are pixels with similar characteristics. For grayscale images this means similar pixel intensity, similar brightness. Note, these regions are not required to be adjacent. The algorithm checks neighboring regions or pixels to see if they are similar enough to be considered of the same group. If the neighboring regions are distinct enough then the region stops growing and the segmentation lines stay up. Regions with same the intensity, no matter their location in the image, are grouped together. Once all the regions can no longer grow, because neighboring regions are too distinct, the segmentation is finished (Chityala, 2014, pg. 190). Watershed groups the foreground into different groups depending on their pixel intensity. In other words the foreground is separated into several groups depending on their lighting. Thus, Watershed segmentation is useful for separating the foreground into smaller groups. However, Watershed works primarily for images with a uniform foreground and uniform background. Variance in the background or foreground or both will cause Watershed to confuse part of the foreground as the background. In the visual glossary in another example of Watershed Segmentation.

Chosen Segmentation

Otsu segmentation efficiently separates the fingers from the guitar since the two areas are distinct enough from one another. This means the test images' histograms have two well-defined peaks and a valley. Adaptive Thresholding also separates the fingers from the guitar but using local thresholds causes the algorithm to treat parts of the guitar as the foreground. Although the fingers are detected, the background should not be the same color as the fingers. This segmentation does not allow the outputted image to differentiate the background from the fingers. Similarly, Watershed also struggles with segmenting the fingers. The images have similar pixel intensity and shading throughout the images resulting in the foreground mixed up with the background. Parts of the fingers were mixed in with the guitar which does not lend itself well for these segmenting.

Quantifying the Difference

After subtraction and segmentation the images will appear like the images in the visual glossary on page thirty-eight. Each segmented image has the fingers in white and the background in black. This means the pixels representing the difference have a value of 255 (white) and the background has a value of zero (black.) Quantization of the difference between a test image and a control image is possible by counting the white pixels in the image to measure the difference between the pictures. All the white pixels are counted and divided by the total amount of pixels in the image. The result is a percentage called Rate of Difference. On top of each picture on page thirty-eight their rate of difference (ROD) in percentage. ROD shows how much each picture was different from a correctly played G-chord.

Chapter 5

Results

There is a distinct difference between the bad G-chords and good G-chords. From our sample size the average ROD for incorrect played G-chords is 27.622%, whereas the average ROD for correctly played G-chords is 11.362%. The algorithm generally detects more difference in badly played G-chords. However, we must acknowledge there is only a 10% difference on average between a bad G-chord and good G-chord. The smallest difference for bad G-chord was 23.7% and the highest difference for a good G-chord was 22.99%. Thus, there is less than one percent difference (0.71%) between these images.

Although this paper demonstrates how image processing can distinguish a good G-chord from a bad G-chord, the main limitation of this algorithm is the usage of a control image. As the control image changes so will the rate of difference for each test image. Universally defining a good G-chord without depending on a control image is needed to improve the accuracy of recognizing good chords. A universal G-chord should focus on the most important positions of a guitar chord which are the tips of the fingers press onto the guitar. This software compares the entire image of chord as opposed to just the parts which make a G-chord sound correctly. Focusing on smaller areas will minimize unwanted data.

Another limitation is the slow speed and unstable nature of data collection. Taking pictures takes too long and many things can be different between test images such as lighting and positioning of the camera. These properties impact the accuracy of the data collection. The algorithm can be much more precise by improving the data collection through video input. Video input increases the amount of images being processed and provides live feedback to the user.

Future Work

Having the software work in real time is the largest requirement for this technology to be beneficial. Meaning this paper will be greatly improved with a GUI and video input. A Graphics User Interface will make or break the technology for many users, especially since our focus group requires a GUI which attends to the hard-of-seeing. In order for the software to work in real time video input is required. Setting up a camera to take video of the guitarist will provide constant data collection, and constantly be checking for the right chord. Since the data is constantly collected this will to minimize difference in lighting and rotation. Having a stand for the camera will help stabilize the data collection. Furthermore, the product needs be easily set up for the hard-of-seeing guitarists. The product needs to recognize a guitar to help the users correctly face the camera toward themselves.

Another progression would be to detect a sequence of chords. Most guitar songs with chords have a repeatable pattern. It might be beneficial to have to program detect chord progression as the guitar begins expanding their knowledge.

One more expansion for the software is an audio component. Visual software in combination with sound software. Many products and app detect sounds like tuning guitars. Recognizing different chords by sight and sound provides more data about the aspiring guitarist.

References

Arky, B. (2014, April 06). Understanding Visual Processing Issues. Retrieved September 19, 2016, from https://www.understood.org/en/learning-attention-issues/child-learning-disabilities/visual-processing-issues/understanding-visual-processing-issues

Assistive Devices for People with Hearing, Voice, Speech, or Language Disorders | NIDCD. (2014, July 03). Retrieved September 19, 2016, from https://www.nidcd.nih.gov/health/assistive-devices-people-hearing-voice-speech-or-language-disorders

Chityala, R., & Pudipeddi, S. (2014). *Image processing and acquisition using Python*. Boca Raton, FL: Taylor & Francis.

Dreon, O., Kerper, R. M., & Landis, J. (2011). Digital Storytelling: A Tool for Teaching and Learning in the YouTube Generation. Middle School Journal, 42(5), 4-9.

Geometric Image Transformations. (n.d.). Retrieved November 14, 2017, from http://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html#getperspectivetransform

Godsey, M. (2015, March 25). The Deconstruction of the K-12 Teacher. Retrieved October 18, 2016, from http://www.theatlantic.com/education/archive/2015/03/the-deconstruction-of-the-k-12-teacher/388631/

Jones, P. (2007, April). Music Education for Society's Sake: Music Education in an Era of Global Neo-Imperial/Neo-Medieval Market-Driven Paradigms and Structures. Action, Criticism, and Theory for Music Education, 6(1). Retrieved August 15, 2016, from http://act.maydaygroup.org/articles/Jones6_1.pdf

Knight, D., Tang, M., Dahlkamp, H., & Plagemann, C. (2010). *A Framework for Recognizing Hand Gestures* [Scholarly project]. In *CiteSeerx*. Retrieved March 31, 2016, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.374.8150&rep=rep1&type=pdf

Nielsen, E. (n.d.). The Impact of the Digital Age on Professional Musicians. Retrieved September 02, 2016, from http://www.olafurandri.com/nyti/papers2013/The Impact Of The Digital Age on Professional Musicians.pdf

Pramada, S., Saylee, D., Pranita, N., Samiksha, N., & Vaidya, A. S., Mrs. (2013). Intelligent Sign Language Recognition Using Image Processing. *IOSR Journal of Engineering, 3*(2), 2nd ser., 45-51. Retrieved March 05, 2016, from http://www.iosrjen.org/Papers/vol3_issue2 (part-2)/H03224551.pdf

Pring, L., & Ockelford, A. (2005). Children with septo-optic dysplasia – musical interests, abilities and provision: The results of a parental survey. The British Journal of Visual Impairment, 23(2), 58-66. Retrieved September 21, 2016.

Qing, L. (2007). Student and Teacher Views About Technology: A Tale of Two Cities?. Journal Of Research On Technology In Education, 39(4), 377-397.

RNIB. Teaching science to students with vision impairment. (2015, November 15). Retrieved October 17, 2016, from http://www.rnib.org.uk/insight-online/teaching-science-students-vision-impairment

Rokade, R. S., & Doye, D. D. (2015). Spelled sign word recognition using key frame. *IET Image Processing*, *9*(5), 381-388. doi:10.1049/iet-ipr.2012.0691

Rosebrock, A. (2015, July 13). 4 Point OpenCV getPerspective Transform Example. Retrieved September 07, 2017, from http://www.pyimagesearch.com/2014/08/25/4-point-opencv-getperspective-transform-example/

Stafford, S. A. (2010, Fall). Music in the Digital Age: The Emergence of Digital Music and Its Repercussions on the Music Industry. The Elon Journal of Undergraduate Research in Communications, 1(2), 112-120. Retrieved September 17, 2016, from http://www.elon.edu/docs/e-web/academics/communications/research/vol1no2/09staffordejfall10.pdf

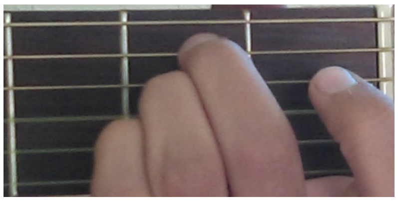
The ImageChops Module. (n.d.). Retrieved January 14, 2017, from http://effbot.org/imagingbook/imagechops.htm#tag-ImageChops.difference

Willings, C. (n.d.). Assistive Technology for Students who are Blind or Visually Impaired. Retrieved September 19, 2016, from http://www.teachingvisuallyimpaired.com/assistive-technology.html

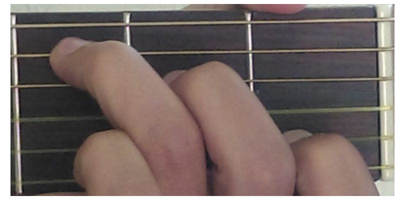
Yuan, B, Folmer, E. Blind hero: enabling guitar hero for the visually impaired, Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility, October 13-15, 2008, Halifax, Nova Scotia, Canada [doi>10.1145/1414471.1414503]

Before Warp Pictures

bad_G_1



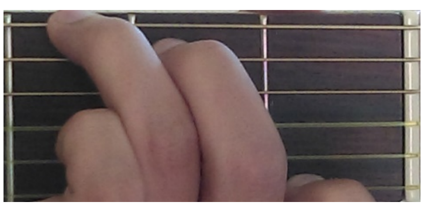
bad_G_2



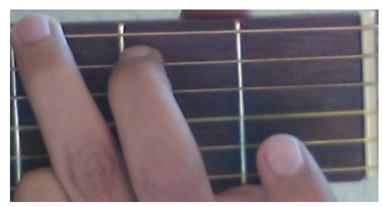
bad_G_3

bad_G_4

bad_G_5

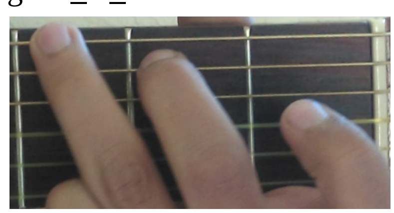


good_G_1

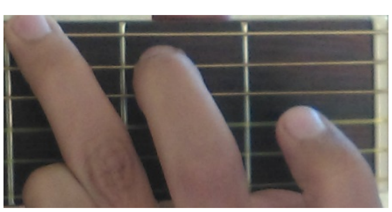


good_G_2

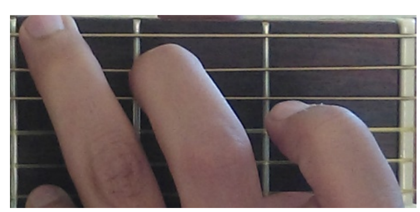
good_G_3



good_G_4

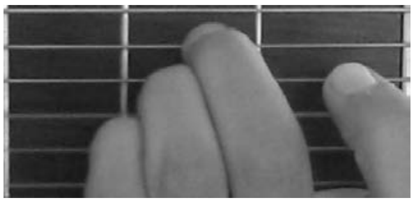


orignial_G



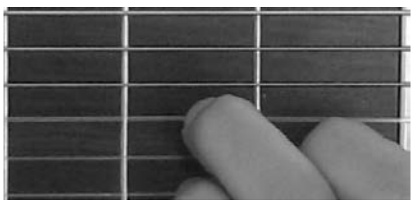
After Warp Pictures

bad_G_1_Final

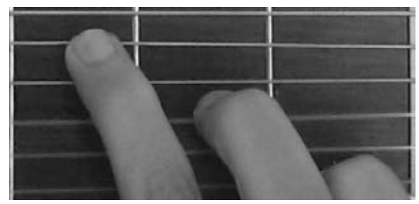


bad_G_2_Final

bad_G_3_Final

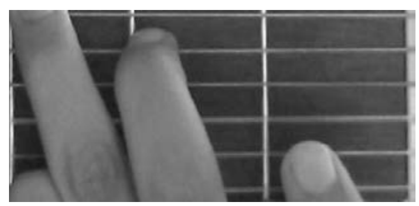


bad_G_4_Final



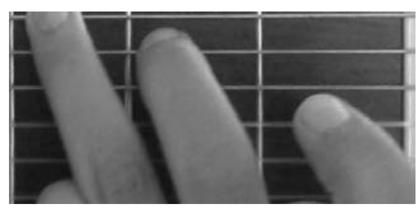
bad_G_5_Final

good_G_1_Final



good_G_2_Final

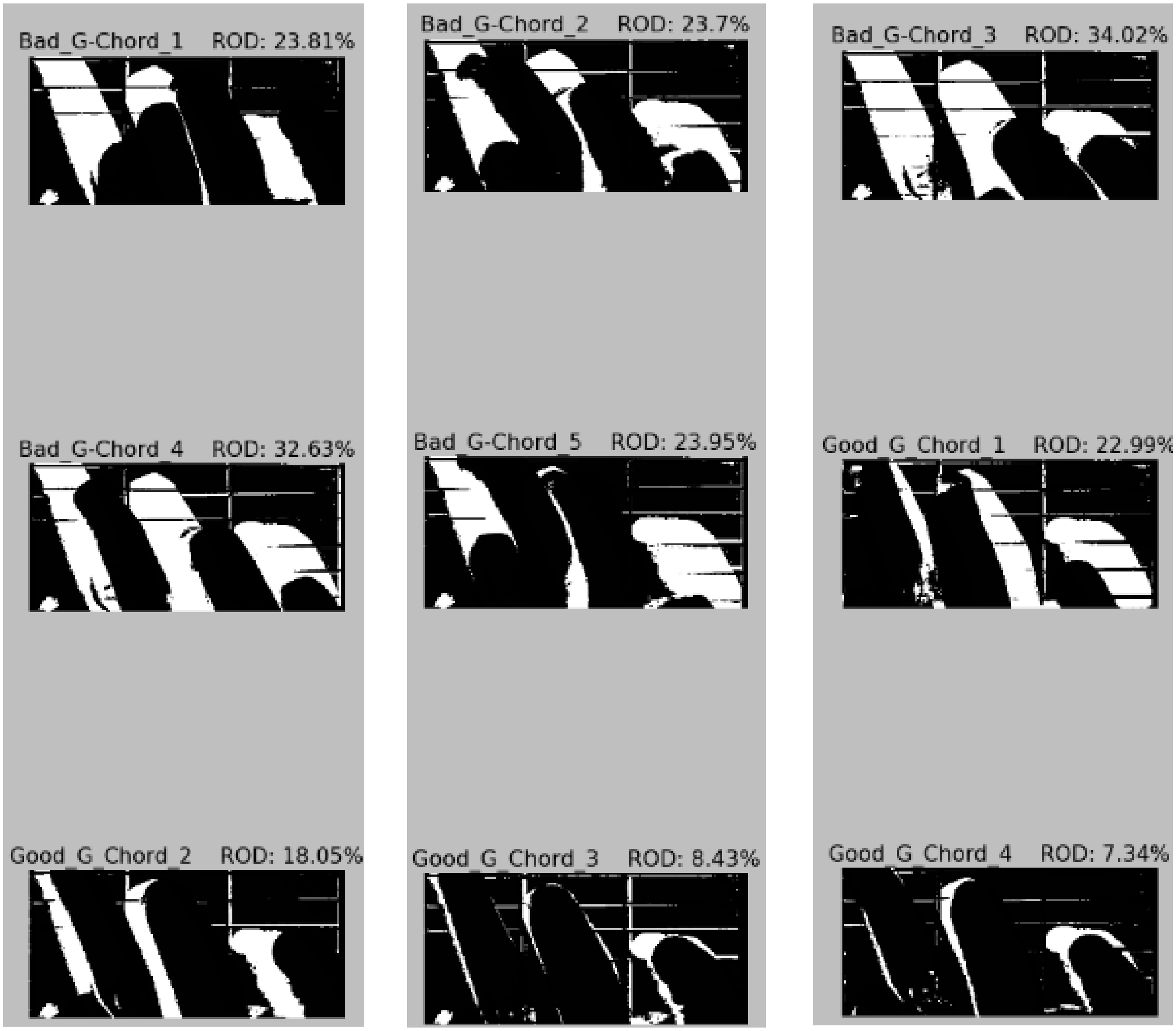
good_G_3_FIinal



good_G_4_Final

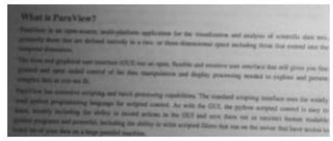
orignial_G_Final

Results

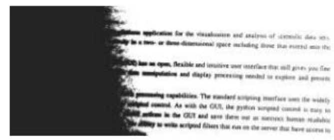

Bad_G-Chord_1 ROD: 23.81%
Bad_G-Chord_2 ROD: 23.7%
Bad_G-Chord_3 ROD: 34.02%
Bad_G-Chord_4 ROD: 32.63%
Bad_G-Chord_5 ROD: 23.95%
Good_G_Chord_1 ROD: 22.99%
Good_G_Chord_2 ROD: 18.05%
Good_G_Chord_3 ROD: 8.43%
Good_G_Chord_4 ROD: 7.34%

Visual Glossary

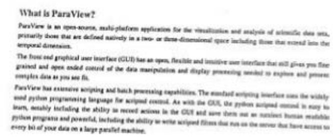
Adaptive Thresholding versus Otsu's method



(a) Input image.



(b) Output using Otsu's method.



(c) Output using adaptive thresh-olding.

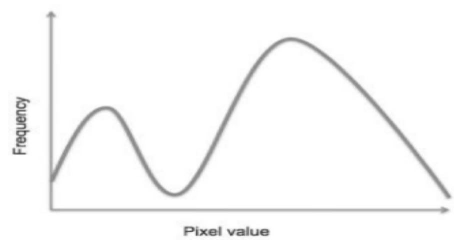
Chitvala, 2014

FIGURE 7.5: An example of thresholding with adaptive vs. Otsu's.

GetPerspecticeTransform()

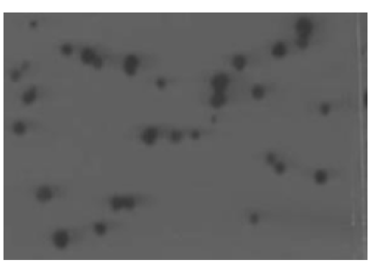
```
pts1 = np.float32([[10,2],[510,4],[3,238],[505,234]])

pts2 = np.float32([[0,0],[512,0],[0,242],[512,242]])

pt   = cv2.getPerspectiveTransform(pts1,pts2)
```

Histogram



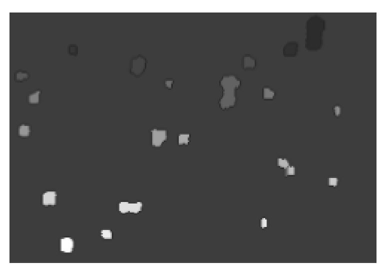
Chityala, 2014

Otsu segmentation v. Watershed segmentation



(a) Input image.

(b) Thresholded image using Otsu's.



(f) Output

Chityala, 2014

Warp Perspective Transform

```
import numpy as np
import PIL.ImageOps as ops
import PIL.Image as image
import scipy.misc as misc
import cv2
from matplotlib import pyplot as plt

original = ops.grayscale(image.open(“”).resize((512,242)))
# Resizes the image to a 512 x 242 and grayscales the image

beforeTransform = original
# Use to reference the orignial version of the image
after = misc.fromimage(original)
# Convert the image into a ndarray

pts1 = np.float32([[10,2],[510,4],[3,238],[505,234]])
# 4 Coordinate points from original image to become new corners; TL, TR, BL, BR

pts2 = np.float32([[0,0],[512,0],[0,242],[512,242]])
# These four coordiante points will become the corners of the image; TL, TR, BL, BR

pt = cv2.getPerspectiveTransform(pts1,pts2)
# Get the transformation matrix

afterTrans = cv2.warpPerspective(after,pt,(512,242))
# Apply the transformation to orignial image. Set final size of image to 512 x 242

afterTrans = misc.toimage(afterTrans)
# Convert the new array into an image

#orignTrans.save('')
# Once the user is content with transformed image uncomment the line to save the image

plt.subplot(121), plt.imshow(afterTrans,cmap=plt.cm.gray),plt.title('After')
#Subplot for after the transformation

plt.subplot(122), plt.imshow(beforeTransform,cmap=plt.cm.gray),plt.title('Before')
#Subplot for before the transformation

plt.yticks(np.arange(0,242,10))
# Tick in x direction every 10 steps in order to improve accuracy for guessing coordinates
plt.xticks(np.arange(0,512,25))
# Tick in y direction every 10 steps in order to improve accuracy for guessing coordinates
plt.show()
# Show plot
```

Global Segmentation Subtraction

```
import PIL.ImageChops as chops
import PIL.Image as image
import scipy.misc as misc
from matplotlib import pyplot as plt
from matplotlib import gridspec
from skimage.filter import threshold_otsu

def OtsuSegmentation(img1):
	o = img1.convert('L')
	o = misc.fromimage(o)
	thresh = threshold_otsu(o)
	otsu = o > thresh
	otsu = misc.toimage(otsu)
	return otsu

def compareImages(img1, img2):
	sub = chops.subtract(img1,img2,scale=1.0,offset=1)
	pixels = list(sub.getdata())
	diffPix = 0.0
	totPix = float(len(pixels))
	for pix in pixels:
		if pix == 255:
			diffPix = diffPix+1;
	diffPix = diffPix/totPix*100
	diffPix = round(diffPix,2)
	return diffPix, sub

base = OtsuSegmentation(image.open('orignial_G_Final.jpg'))

bad1 = OtsuSegmentation(image.open('bad_G_1_Final.jpg'))
bad2 = OtsuSegmentation(image.open('bad_G_2_Final.jpg'))
bad3 = OtsuSegmentation(image.open('bad_G_3_Final.jpg'))
bad4 = OtsuSegmentation(image.open('bad_G_4_Final.jpg'))
bad5 = OtsuSegmentation(image.open('bad_G_5_Final.jpg'))

good1 = OtsuSegmentation(image.open('good_G_1_Final.jpg'))
good2 = OtsuSegmentation(image.open('good_G_2_Final.jpg'))
good3 = OtsuSegmentation(image.open('good_G_3_Final.jpg'))
good4 = OtsuSegmentation(image.open('good_G_4_Final.jpg'))

testImages = [bad1,bad2,bad3,bad4,bad5,good1,good2,good3,good4]
gs = gridspec.GridSpec(3,3)
diffRates = []
```

```
for pos,testImage in enumerate(testImages):
	diffRate, subtracted = compareImages(base,testImage)
	if pos == 0:
		bd1 = plt.subplot(gs[0,0])
		bd1.imshow(subtracted,cmap=plt.cm.gray), bd1.axis('off'),
		bd1.set_title('Bad_G-Chord_1 ROD: ' + str(diffRate) + '%')
	if pos == 1:
		bd2 = plt.subplot(gs[0,1])
		bd2.imshow(subtracted,cmap=plt.cm.gray),
		bd2.axis('off'), bd2.set_title('Bad_G-Chord_2 ROD: ' + str(diffRate) + '%')
	if pos == 2:
		bd3 = plt.subplot(gs[0,2])
		bd3.imshow(subtracted,cmap=plt.cm.gray),
		bd3.axis('off'), bd3.set_title('Bad_G-Chord_3 ROD: ' + str(diffRate) + '%')
	if pos == 3:
		bd4 = plt.subplot(gs[1,0])
		bd4.imshow(subtracted,cmap=plt.cm.gray),
		bd4.axis('off'), bd4.set_title('Bad_G-Chord_4 ROD: ' + str(diffRate) + '%')
	if pos == 4:
		bd5 = plt.subplot(gs[1,1])
		bd5.imshow(subtracted,cmap=plt.cm.gray),
		bd5.axis('off'), bd5.set_title('Bad_G-Chord_5 ROD: ' + str(diffRate) + '%')
	if pos == 5:
		gd1 = plt.subplot(gs[1,2])
		gd1.imshow(subtracted,cmap=plt.cm.gray), gd1.axis('off'),
		gd1.set_title('Good_G_Chord_1 ROD: ' + str(diffRate) + '%')
	if pos == 6:
		gd2 = plt.subplot(gs[2,0])
		gd2.imshow(subtracted,cmap=plt.cm.gray), gd2.axis('off'),
		gd2.set_title('Good_G_Chord_2 ROD: ' + str(diffRate) + '%')
	if pos == 7:
		gd3 = plt.subplot(gs[2,1])
		d3.imshow(subtracted,cmap=plt.cm.gray), gd3.axis('off'),
		gd3.set_title('Good_G_Chord_3 ROD: ' + str(diffRate) + '%')
	if pos == 8:
		gd4 = plt.subplot(gs[2,2])
		gd4.imshow(subtracted,cmap=plt.cm.gray), gd4.axis('off'),
		gd4.set_title('Good_G_Chord_4 ROD: ' + str(diffRate) + '%')
	diffRates.append(diffRate)
plt.tight_layout()
plt.show()
```