**The Combination of Noise Deduction, Image segmentation and Image sonification**

**Zi Wang, Tony Huang**

***Abstract-* This research paper provides a thorough analysis of the complex of image noise, median filter, algorithms for image segmentation include edge deduction, K-mean and image sonification include introduce a useful tool for musical implementation which is JypthonMusic. The study primarily focuses on salt and pepper image noise solve the problem of how to add salt paper noises into a PPM (Portable Pixel Map)format image with different percentages, and how to deduct those added salt paper noise with median filter algorithm, and how to do edge detection and k-mean and how to do image preprocess for image sonification*.***

**Introduction**

We use C++ to represent the implementation of add salt pepper noise and removal by median filter, then implement image segmentation based on Edge detection and K-Means clustering, finally base on image segmentation result we implement image sonification . Our input is a colorful PPM image with a size of 1685KB. First. we create a function to read and write PPM images. Then uniform randomly assign random pixel value to random pixels to complete noise add on. We create median filter to average pixel value to complete noise removal. in edge detection, we use a simple average to transform a PPM color image into a grayscale image and then use the Sobel method to find the edge by calculating the pixel values from the grayscale image. For the K-means clustering, we decide to pick 5,10,15 cluster centers(K numbers) to show the best result of our input image. First, we randomly select K data points, and then we calculate the distance between each data point and cluster centers. After that, we label each data point to each cluster, and then we calculate the Sum Square Error (SSE) to find the new centers. The program we write will repeat the process until the SSE of new centers are equal to the previous one.

The output of image noise removal will include image with noise and image after median filter, the output of image segmentation project will create 3 separate images including the output of the K-means clustering, the edge deduction of the original image, and the edge deduction base on K-means output. Finally, the output of sonification will be an audio base on input image. From the implement the image segmentation of Edge detection and K-Means clustering, we can clearly see the difference between the original image, the image after the edge deduction process, and the image after the k-mean clustering process. The image sonification will play different sounds base on different input image and we can tell the difference base on different K number of clustering.

**PPM file Introduction**

"PPM" is an acronym derived from "portable pixel map" [1]. Images in this format are also known as "portable pix maps". The PPM format is the lowest common denominator color image file format. Although this format is very inefficient. It is highly redundant and contains a lot of information that the human eye cannot even discern. Additionally, the format only allows very little image information beyond basic colors, which means that files in this format may have to be combined with other independent information to take full advantage of it. However, it is very easy to write and analyze programs to handle this format.

**Salt and pepper noise**

Salt and pepper noise is a type of impulse noise that appears as randomly scattered black or white pixels in an image. This noise is caused by errors in data transmission, memory cell failures, or issues with image digitizing or transmission. In an 8-bit image, salt and pepper noise randomly occurs in certain pixels, resulting in extreme values of either 0 or 255. [9] The noise significantly damages image information, making it difficult to perform tasks such as edge detection or image segmentation and recognition. This type of noise is also known as spike or fat-tailed distributed noise because it is caused by sudden and sharp disruptions in the image signal. To reduce salt and pepper noise, common methods include using median or morphological filters. Typically, the intensity value for pepper noise is close to 0, while for salt noise, it is close to 255. [10] Only a proportion of image pixels are affected, and the remaining unaffected pixels remain unchanged.

**The Implementation of SALT-AND-PEPPER NOISE**

During the implementation, we add different percentages of salt pepper noises by using a uniform random algorithm. The process steps of this part include:

1. The program reads image information and then stores it in image structure.

2. Calculate total data points. We calculate the total data points by multiple the size of the x/y axis from the sample image.

3. After we get the total number of data points, we pick the percentage of noise that we plan to add to the image. The number of noises is calculated by multiplying the percentage of noise and the total data point.

4. The next step is to set up the pixel value of noise, for this implementation, we uniformly randomly pick the range between 0 to 255.

5. After that, we need to randomly add those noises into the image. We store the RGB value for every data point from images into three separate vectors which means those vectors represent the whole pixel information of our sample image. We also use uniform random to pick data points from those vectors and use noises’ pixel value to replace the original value. That is the process of how to add salt pepper noises into the original image. After we complete those steps, we just need to write out the noised image and save it for the following up median filter implementation.

**Median filter method**

The median filter is a widely used nonlinear filtering method for removing impulse noise such as salt and pepper noise from digital images. [11] It works by establishing a window of size n\*n and replacing the pixel value of the processing pixel with the median value of the neighboring pixels within the window. The median value is calculated by first sorting all the pixel values from the window and then selecting the middle pixel value. This order-statistics filter, also known as the Median filter, is particularly effective in preserving sharp edges in images, making it more robust than mean filters. [12] It is effective for images with low noise contamination density, but as the density increases, the window size needs to be enlarged to ensure the removal effect. However, this can lead to blurred edges and details. Compared to linear smoothing filters of similar size, median filters provide excellent noise reduction capabilities with less clouding. They are suitable for both bipolar and unipolar impulse noise and are often used for speckle noise reduction while preserving edges. Though more complicated, diffusion filters can be more effective for speckle noise reduction. In practice, the implementation of the median filter involves looking at the neighborhood of each element of the signal (image) and selecting the most similar element. This approach makes the median filter an excellent tool for removing salt and pepper noise [13] while preserving important features of digital images.

**The Implementation of the MEDIAN FILTER**

To implement the median filter algorithm, we need to calculate the pixel’s neighbor’s pixel value which will be 8 data points excluding the pixel itself and assuming this pixel is not located in the boarding area. Normally we need to sort those 8 neighbor pixels and calculate the median value. However, for this implementation experiment, we are using the average value of the median and its neighbor, which means the 8-pixel values are sorted from min to max 0 to 7, we use the average of 3 and 4 pixels to implement the median filter method, the process steps include:

1. Read and store image information that is added to salt pepper noises.

2. Store RGB value into 3 separate temp arrays

3. Calculate pixel values by using a nested loop for the x/y axis to calculate the neighbor pixel value. Within this nested loop, we add a sorting loop to calculate the average value of medians. for each index of

x/y axis, there will be 8 pixels to be processed, after that the original pixel value will be replaced by the average value of medians.

4. Repeat step 3 until all original pixels are replaced, then the process is completed.

In this implementation process, we complete the median filter method but slightly different from the original algorithm because we are not using the median of neighbor pixels, instead of that, we are using the average of medians, just for a more clear explanation, for example, after sorting there will be min to max, 0 to 7, the median is 4, we use the average of 4 and 3 (pixel value(3+4)/2) to replace the original pixels. After we complete the median filtering method, we have an output of the image which is noise removed.

**The Implementation of Sobel Edge Detection**

In the process of image segmentation, the most basic step is edge detection. It divides an image into an object and its background. Edge detection divides an image by observing changes in its intensity or pixels. This is the most common method of recovering information from images. There are generally three types of edges: horizontal edges, vertical edges, and diagonal edges [4]. An edge detection technique uses matrix math to count areas of different intensities in an image. In most cases, the edges of objects can be identified by extreme differences in pixel intensity. Finding the edges of a picture is as simple as finding all the large differences in intensity. An image processing algorithm called Sobel edge detection identifies edges on a picture.

**Conversion to Grayscale**

The image is a collection of pixels. Pixels are arranged to form an image. The size of the image is expressed in matrix form as I x J, where “I” is the number of pixels in a row and j is the number of pixels in a column. Each pixel has its pixel value which has three values: red, green, and blue. Grayscale conversion means converting an image to a gray representation, ignoring other colors. This can be done by applying a mathematical formula to the RGB values and replacing them in the image or creating a new image with these new values.

Before doing edge detection, we need to convert the image to a grayscale image. Simple Average performs an average of the RGB values of each pixel. can be calculated as [6]:

Average = (R+G+B)/3

Average = (pixel[i][j][0]+pixel[i][j][1]+pixel[i][j][2])/3

Where i = image length, j = image width, [0], [1], [2] are used to select RGB values.

The example mages below is simple average grayscale conversion:(left) original image (right)grayscale image[6]:

A group of people in a circle

Description automatically generated with low confidence

**Sobel Filter for Edge Detection**

The Sobel edge detection method was proposed by Sobel in 1970.[3] This method detects both horizontal and vertical edges. The main difference between the two methods is that the coefficients of the mask are not fixed. Can be adjusted according to our requirements. The image is first processed separately in the X and Y directions and then combined to form a new image representing the sum of the X and Y edges of the image. The image is then converted from an RGB image to a grayscale image. Then use kernel convolution. The kernel is a 3 x 3 matrix consisting of differently (or symmetrically) weighted indices.

One kernel is simply the other rotated by 90°

Table

Description automatically generated

With kernel convolution, you can see in the example image that there is an edge between the columns with 100 and 200 values.

Calendar

Description automatically generated

image cited from[2].

This kernel convolution is done using the X direction, the image is written from the left scan. If the filter is set to (2,2) in the image above, it has a value of 400, so the point has a rather prominent edge. If you want to exaggerate the edges, you need to change the filter values of -2 and 2 to a higher magnitude. Possibly -5 and 5. This will make the gradient of the edges larger and more pronounced. Once the image has been processed in the X direction, the image can be processed in the Y direction. The magnitudes of the X and Y kernels are then summed to produce a final image showing all edges in the image.

Then the formula to calculate the resulting new p5 pixel is new pixel intensity = (p1+(p2+p2)+p3-p7-(p8+p8)- p9)+(p3+(p6+p6)+p9-p1-(p4+p4)-p7) which is then clamped to the 0-255 range. The actual formula uses the horizontal and vertical components into the final form new pixel intensity = SQRT((X\*X)+(Y\*Y)) where X = (p1+(p2+p2)+p3-p7-(p8+p8)-p9) and Y = (p3+(p6+p6)+p9-p1-(p4+p4)-p7) The Sobel filter is applied to the rest of the image as given below[8].

A picture containing electronics, calculator

Description automatically generated

The calculation of the Sobel operator is very slow, but its larger convolution kernel can smooth the input image to a greater extent, thus making the operator less sensitive to noise.

C++ implementations of Sobel Filter for Edge Detection

Text

Description automatically generated

**Image Segmentation based on Clustering**

clustering is the task of dividing the population of data points into several groups, in the same group, all the data points are more similar and less similar to data points from other groups. In our project, we will use K-means, one of the most used clustering algorithms. K represents the number of clusters which are the centers of each group, for example, if K = 5, we have 5 groups of data points with 5 cluster centers.

here is the process of how K-means clustering works from [7]:

1. randomly select K initial clusters, (random k data points from image)
2. randomly assign each data point to any one of the K clusters.
3. calculate the centers of these clusters.
4. calculate the distance of all the points from the center of each cluster.
5. depending on this distance, the points are reassigned to the nearest cluster
6. calculate the center of newly formed clusters
7. finally, repeat steps 4, 5, and 6 until either the center of the clusters do not change or we reach the set number of iterations

In our project, we use C++ language to represent image segmentation based on K-means. Based on comparing the output of a different number of clusters, we finally pick K=15 to represent the best result.

the function we created can be divided by the following steps:

1. randomly select the first cluster centers. In this part, we randomly select K clusters. which means randomly picking 15 data points from the image, and storing their R, G, and B value in the array.

2. Calculate the distance between each data point and center. we calculate the sum of squared errors (SSE) between them and then label each data point to their closed centers.

3. Get a new center. after we did the first clustering, we find the new cluster centers within each group. we sum all the R, G, and B values and divide them by the number of data points in the cluster. which is the new R, G, B value that is the new cluster center for the group.

4. we calculate the difference between the new cluster centers with the precious ones, if they are not same, we repeat steps 2 and 3. In detail, we calculate the distance between each data point and new centers, get the closed centers, and relabel them into new clusters.

Finally, the new centers are the same as the previous centers, we finish the clustering process, and we write out the image.

**Sonification**

Sonification is a fascinating technique that transforms data into sound, offering a new perspective on data interpretation. It is a complementary approach to data visualization that provides an engaging way of experiencing data. While it is particularly useful for people who are visually impaired, anyone can benefit from sonification as it offers a new way of perceiving and understanding data. Sonification has proven to be especially useful in the field of data analysis. By converting data into sound, patterns, and anomalies that are not easily detected in a visual display can be detected more easily. This has significant implications for those who are visually impaired, as it allows them to access visual data using sound. There are numerous techniques and tools available for sonifying data. One approach involves mapping data values to musical parameters such as pitch, volume, and duration. In the example provided earlier, an image is sonified by mapping pixel luminosity to pitch, pixel redness to duration, and pixel blueness to volume. Sonification is not only limited to data analysis but is also used in scientific research. For instance, in astrophysics, sonification is used to study the sounds of the universe, including the sounds of stars, galaxies, and other astronomical phenomena. To sum up, sonification is an exciting technique that provides an alternative way of representing and understanding data. It offers an engaging and novel way of experiencing data, and its applications extend beyond data analysis, as it can help researchers gain new insights into complex data sets.

For the implementation, we use JythonMusic library. JythonMusic [14]is a tool for creating music and programming that is free and open source. It can be used by anyone, regardless of their level of expertise in music or programming. With JythonMusic, you can make music, manipulate images, create graphical user interfaces, and connect to various external devices like digital pianos, smartphones, and tablets. JythonMusic is built on Python, making it easy for beginners to learn and powerful enough for advanced users. A phrase is a collection of musical notes played one after the other in a sequential manner. Each phrase has a start time, typically represented as a float value (e.g., 0.0 in quarter note time units). If no start time is specified, the phrase will begin at the end of the previous phrase, or the beginning of the piece if it's the first phrase. There are various functions in music libraries to create, retrieve, and modify phrases.  
  
 A Part object is a collection of Phrase objects, which are played together by a single instrument. Even if the instrument does not allow for polyphony, a Part can still have overlapping melodies since the phrases play in parallel. In other words, a Part can be viewed as a group of different instruments of the same type, with each playing a unique melody. A Part can be assigned to sixteen channels available on a computer's audio system. While each channel can play any of the 128 different instruments, only one instrument can be played at a time. channel 9 is reserved for percussion sounds, and if a Part is assigned to this channel, regardless of the instrument, the notes will generate percussion sounds. To create a Part, you can specify a title, the instrument to be used, and the channel to which it will be assigned. For example, p = Part("An example flute part", FLUTE, 0) creates a Part called "An example flute part"; that uses the FLUTE instrument and is assigned to MIDI channel 0.Timer objects allow you to schedule tasks or functions to be executed repeatedly or once, based on a specified time interval.

**Conclusion**

we successfully complete our project which is the Combination of Noise Deduction, Image segmentation and Image sonification. we add noises into a image, then use median filter algorithm to removal noises. Then we implement image segmentation base on algorithm edge detection, image k-mean clustering. We use different cluster numbers to see how cluster numbers affect the image color regions, finally, we use the output from image segmentation to implement image sonification. Different cluster numbers play different sounds.

**Reference**

‌[1]“PPM Format Specification,” *netpbm.sourceforge.net*. https://netpbm.sourceforge.net/doc/ppm.html (accessed Dec. 04, 2022).

[2]“An Implementation of Sobel Edge Detection - Rhea,” *www.projectrhea.org*. <https://www.projectrhea.org/rhea/index.php/An_Implementation_of_Sobel_Edge_Detection>

[3]R. Fisher, S. Perkins, A. Walker, and E. Wolfart, “Feature Detectors - Sobel Edge Detector,” *homepages.inf.ed.ac.uk*, 2003. <https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>

[4]NASIR HUSSAIN DAR,Ramya HR,Dr.V.Thirunavukkarasu,”Image segmentation Techniques and its Applications”

‌[5]V.Gurusamy,[Subbu Kannan](https://www.researchgate.net/profile/Subbu-Kannan),”REVIEW ON IMAGE SEGMENTATION TECHNIQUES”,Oct,2014,

<https://www.researchgate.net/publication/273127438_REVIEW_ON_IMAGE_SEGMENTATION_TECHNIQUES>

[6]samir khanal, “GrayScale Conversion,” *Medium*, Mar. 26, 2020. <https://samirkhanal35.medium.com/grayscale-conversion-56189cd0e9ca> (accessed Dec. 04, 2022).

**‌**[7]Analytics Vidhya, “Step-by-Step Tutorial on Image Segmentation Techniques in Python,” *Analytics Vidhya*, May 06, 2019. <https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/>

[8]“Sobel Edge,” *www.roborealm.com*. <http://www.roborealm.com/help/Sobel.php> (accessed Dec. 04, 2022).

[9] Jayanetti, P. (2020) Remove salt and pepper noise with median filtering, Medium. Analytics Vidhya. Available at: <https://medium.com/analytics-vidhya/remove-salt-and-pepper-noise-with-median-filtering-b739614fe9db>

[10] (PDF) salt and pepper noise: Effects and removal - researchgate (no date). Available at: <https://www.researchgate.net/publication/326408527_Salt_and_Pepper_Noise_Effects_and_Removal>

[11] harati, S. et al. (no date) A comparative analysis of image denoising problem: Noise models, denoising filters and applications, Preprints. Preprints. Available at: <https://www.preprints.org/manuscript/202005.0053/v2>

[12] samir khanal, “GrayScale Conversion,” Laight, S. (2016) Digital Image Processing Second Edition, Optics and Lasers in Engineering. Available at: <https://www.academia.edu/25992484/Digital_Image_Processing_Second_Edition>

[13] <https://samirkhanal35.medium.com/grayscale-conversion-56189cd0e9ca>

[14]JythonMusic https://jythonmusic.me/music-library/

**Output**A collage of a person sitting in a chair

Description automatically generatedA person sitting on a green couch

Description automatically generated with low confidenceA screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated