Saving the Sonorine: Photovisual Audio Recovery Using Image Processing and Computer Vision Techniques

Kai Ji (Kevin) Feng ’21
kjfeng@princeton.edu

Advisor: Adam Finkelstein
af@cs.princeton.edu

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Engineering Department of Computer Science Princeton University

January 7, 2020
Honor Statement

I hereby declare that this Independent Work report represents my own work in accordance with University regulations.

Kai Ji (Kevin) Feng ’21
Saving the Sonorine: Photovisual Audio Recovery Using Image Processing and Computer Vision Techniques

Kai Ji (Kevin) Feng ’21
kjfeng@princeton.edu

Abstract

This paper presents a novel technique to recover audio from sonorines, an early 20th century form of analogue sound storage. Our method uses high resolution photographs of sonorines under different lighting conditions to observe the change in reflection behavior of the physical surface features and create a three-dimensional height map of the surface. Sound can then be extracted using height information within the surface’s grooves, mimicking a physical stylus on a phonograph. Unlike traditional playback methods, our method has the advantage of being contactless: the medium will not incur damage and wear from being played repeatedly. We compare the results of our technique to a previously successful contactless method using flatbed scans of the sonorines, and conclude with future research that can be applied to this photovisual approach to audio recovery.
Contents

1 Introduction .................................................. 4

2 Related Work .................................................. 5
   2.1 Previous Work Outside of the Project ........................... 5
   2.2 Previous Work on the Project .................................. 7

3 Approach ..................................................... 8

4 Implementation ............................................... 10
   4.1 Photometric Stereo Surface Normal Map Computation .......... 10
   4.2 Calculating Light Direction in Photo ................................ 11
      4.2.1 Selecting the Medoid ...................................... 13
   4.3 Image Processing .............................................. 13
      4.3.1 Tackling Light Variance Over the Sonorine ................. 14
      4.3.2 Improving the Normalization Algorithm .................... 14
   4.4 Photometric Stereo Verification .................................. 17
      4.4.1 RGB Method ............................................... 18
      4.4.2 HSV Method ............................................... 18

5 Evaluation .................................................... 19
   5.1 Normal Map Results .......................................... 19
   5.2 Height Map Results .......................................... 20
   5.3 Sound Results ............................................... 23

6 Discussion and Future Work .................................. 25
   6.1 Discussion of Results ...................................... 25
   6.2 Future Work ................................................. 26

7 Acknowledgements ............................................. 27
1 Introduction

Sonorines are an analogue sound storage medium, comprised of physical inscriptions of recorded audio on an impressionable material, such as wax, spread thinly on card stock. They can be found on the backs of special early 20th century postcards, known as phonocards. A phonocard can then be played with an apparatus known as a phonopostal, which acts like a small handheld gramophone: it allows users to place the phonocard into a designated rotating platform, and a probe would register and afterwards reproduce the recorded audio message [1]. Sonorines were preceded by wax cylinders and succeeded by phonographic records [2]. Interestingly, they are used almost exclusively for recording human voices. One sonorine can usually hold around 75 to 80 words, which suffices for news and voice memos [1].

Jules Verne, a French playwright, novelist, and poet, was credited for conceiving the idea of the sonorine when he replaced the wax disk he used to store sound engravings with a piece of textured paper, which would allow him to mail the medium like a letter [1]. The main merit of the invention was its simplicity: sonorines can be produced by an ordinary, compact phonopostal by the means of a stylus provided with a sapphire point. The point presses on an impressionable substance spread across the surface of the card, carving a spiral that commences on the outside edge of the card and continues in an ever-narrowing fashion until it forms a circle around the size of a small coin near the card’s center. The result is a portable medium with all the advantages of a bulky wax cylinder. Sonorines are able to withstand the potential damage of transmission by mail: the grooves of sonorines are deep enough so that usually no more than two or three syllables are lost in the event that the post office places part of their stamps on the tightly spiraling lines [1].

Although the concept was initially an exciting one, sonorines, along with phonocards and phonopostals, never really took off and were only popular primarily in France and Germany from about 1905 to 1907 [3]. Some called it a “resounding technological failure”. Others considered its invention to be simultaneously late and premature to the recording and communication sphere due to the growing popularity of the telephone amidst little to no changes in expression through written communication [2]. The experimental medium was still nevertheless widely considered in Europe a testament to the innovative and creative spirit of the French [2].

The German department at Princeton has a large collection of phonocards in relatively good condition and would like to recover the audio from within the sonorines to study the contents. Since sonorines were mainly used to record human voice, they can contain valuable information regarding the news, events, and culture of the time. However, conventional tools of sonorine audio playback such as phonopostals are quite rare, as the medium was only popular for a couple of years. Furthermore, even if a phonopostal is successfully located and under Princeton’s possession, the device is over 100 years old and there is no guarantee that the internal mechanics will function properly for effective sound recovery.
In an effort to tackle these challenges, Professor Thomas Levin of the German department and Professor Adam Finkelstein of the Computer Science department have established a collaborative effort to develop a novel optical capture process that recovers audio from sonorines without the need for physical contact. The project started in 2017 and is primarily funded through a two-year grant from the David A. Gardner ’69 Magic Project in the Humanities Council, and will receive support from the DFR Innovation Fund to build a device that combines custom software with off-the-shelf scanning hardware to transform the cards’ audio data into sound [3]. Over the past two years, faculty, students and collaborators at Princeton have taken different approaches to this problem. Recently, one method was able to successfully recover recognizable, albeit noisy, human voice from some sonorines by imaging the sonorines with a flatbed scanner and then running some computations on the images.

The primary goal of this paper is to develop a novel method of audio recovery from sonorines using photographs taken in Firestone Library instead of the previously used flatbed scanner. Since the photos contain richer lighting information and are of a higher fidelity than the uniformly lit flatbed scans, it may allow for higher quality playback. Once we are able to extract the sound from the photos, we can compare the recordings both qualitatively and quantitatively and determine whether or not the photo-based graphic approach is worth adapting. The secondary goal is to develop a set of image processing algorithms that would make future investigation and preservation of sonorines more accessible and efficient.

2 Related Work

2.1 Previous Work Outside of the Project

To our knowledge, there is no record of image-based sonorine audio recovery attempts outside of this project at Princeton, but there are numerous existing works that tackle similar problems and utilize techniques that have been used to inform the approach in this paper.

A 2009 paper by William Clarkson et al. [4] presents a novel method for authenticating physical documents based on random, naturally occurring imperfections in paper texture. The method measures the three-dimensional surface of a page using only a commodity scanner and without modifying the document in any way. To capture the document surface texture, the paper incorporates a computer vision technique called photometric stereo, first introduced by Robert Woodham in a 1992 paper entitled Photometric Method for Determining Surface Orientation from Multiple Images [5]. The document is scanned at varying orientations: 0°, 90°, 180°, and 270°, and the four images under varying angles of illumination can be used to reconstruct the object’s surface normal vectors. This is due to the fact that the light reflected by a surface is dependent on the orientation of the surface in relation to the light source and the observer, so given a sufficient number of light sources under
different angles, the surface orientation can be constrained to a single vector per pixel. However, traditional photometric stereo assumes a point light source, whereas scanners contain a linear one [5]. The paper derives a novel photometric stereo solution for flatbed scanners that does not require extensive calibration of the surface used by previously known methods. By using the page’s physical features, a concise fingerprint can be generated that uniquely identifies the document. The technique is secure against counterfeiting and robust to harsh handling; it can be used even before any content is printed on a page. It has a wide range of applications, including detecting forged currency and tickets, authenticating passports, and halting counterfeit goods. The physical texture of the paper can also be an excellent resource when retrieving information about the direction of light striking the surface, as seen in a paper from the University of Oxford [6]. The paper was able to estimate the source light vector in 3D space from a photo of a textured surface using the observed reflectance and shadow behavior. With regards to our paper, the same photometric stereo approach using light directions computed from images is key to uniquely capturing surface normals of the sonorines.

In the audio domain, efforts of image-based audio playback from analog sound storage media date back to as early as 2003, to a paper by Carl Haber and Vitaliy Fadeyev [7] that marked the beginnings of IRENE (Image Reconstruct Erase Noise Etc.) digital imaging technology. The IRENE system uses a high-powered confocal microscope to follow the path within the groove of a disk or cylinder as the object rotates underneath. Since the texture within the groove dictates audio playback, detailed images of the audio information can be obtained. Depending on how the grooves were cut when the audio was recorded, the system may use different lighting strategies or tracking lasers to ensure the grooves’ visibility to the camera. Custom software then processes the resulting images, converting the texture within the grooves into a digital audio file.

One advantage of the system over traditional stylus playback that is desirable in our project is that it does not require contact with the audio carrier, and so avoids damaging or wearing out the grooves during playback. Other advantages include allowing for the reconstruction of broken or damaged media such as cracked cylinders or delaminating lacquer discs, which cannot be played smoothly with a stylus. Many skips or damaged areas repaired by IRENE eliminates much of the noises that would be created by stylus playback. However, it may also result in the reproduction of more noise, as imperfections in the groove can often be more finely captured than with a stylus. In 2005, Haber and Fadeyev built physical IRENE machines to streamline the playback process by providing all the necessary hardware and software in one place. As of 2019, IRENE machines are operated by three institutions: Lawrence Berkeley National Laboratory, the Library of Congress, and the Northeast Document Conservation Center (NEDCC) [8]. In an attempt to extract audio from sonorines at Princeton, a few of them were sent to NEDCC but the operation was quite expensive, and resulting audio clips were laden with noise. The desire for a cheaper, higher quality solution was in fact one of the motivators for our project.

Around the same time, a similar project was ongoing in Switzerland, stemming from a paper from Ottar Johnsen et al. from the University of Fribourg, entitled Detection of the
**Groove Position in Phonographic Images** [9]. The paper proposes a three-step process to capture sound from images of phonographic records called VisualAudio. First, a picture of a disk is taken with an analogue camera to preserve the sound information in case the original record deteriorates. Then, when one wants to recover the audio, the film is digitized using a specially designed rotating glass turntable. During scanning, the semi-transparent film lies on the glass illuminated from a light source below and a linear camera with a Charge-Coupled Device (CCD) sensor mounted on microscope optics is fixed above the glass. During each rotation of the turntable, one ring of the film is scanned, with the width dependent on optics magnification. By radially displacing the tray, adjacent rings are scanned in order to digitize the whole record.

The circular scanner has the advantage of transforming the circular disk picture into a rectangular image, thus avoiding a coordinate transformation. The image of the whole record is a matrix, where one axis corresponds to the radial position, and the other axis to time. The sound is then extracted from the digitized image using the grooves’ radial position data. Since the sound is not contained in the radial groove positions themselves but the radial velocity, the derivatives of the positions are computed. A lowpass filter is then applied to suppress some of the band noise associated with the records’ limited bandwidth, to get the resulting sound. Although the image scanning techniques from the VisualAudio project require sophisticated procedures and equipment [10] that makes it not so practical for our project, the sound extraction technique was an inspiration in developing the one we currently use.

### 2.2 Previous Work on the Project

This project has been ongoing at Princeton since 2017, and much has been done before we started to investigate our photo-based approach for sound recovery. A relatively successful effort that completed recently involved scanning the sonorines with a flatbed scanner, using photometric stereo to recover the height field of the sonorine, and processing the height map to extract the audio.

A couple of undergraduate students contributed heavily to the scanner variant of this project during the summer of 2019. Rohit Narayanan ’22 developed a mathematical approach that takes as input processed images of the same object from different lighting angles, calculates the surface normal at each pixel using photometric stereo, and uses the surface normals with matrix algebra to compute the height map. For the past couple of summers, Ezra Edelman ’23 developed a program to translate a height map of a sonorine into an audio file. The program mimics a phonopostal’s stylus by estimating the center of the sonorine and tracing the grooves from the inside out, using the height data’s derivative as well as a user-defined sampling frequency to restore the sonorine’s audio.

Human voice can be recognized from the audio of the flatbed scans, but the fidelity of the scans is still inferior to images taken from a sophisticated digital camera. There is a sizable
collection of sonorine photos taken from a camera in Firestone Library: all the photos are taken from directly above the sonorine and each sonorine has four different photos taken with four different lighting angles. Photos of blank paper and a 4x4 grid of mirror spheres were also taken from these configurations for the purposes of calibration and light direction detection. Our project uses these images instead of the flatbed scans as a starting point in hopes of obtaining higher quality audio.

Mariah Crawford ’22 spent the past summer developing a GUI to compute the direction of light on a surface using highlights on the mirror spheres. The GUI detects the position of the mirror spheres as well as the locations of the highlights on the spheres and allows the user to manually adjust any errors. After the user confirms the coordinates, the GUI will calculate light vectors for each of the spheres by estimating their shadow locations with trigonometry and vector algebra. Although these calculations were not accurate enough to be used in our paper, the GUI does manage to retrieve some important data that allows us to develop our own method for computing light direction.

3 Approach

We already have some tools built for the flatbed scanner variant of this project that we can use in this paper with slight modification, namely the height map generation techniques and the height-to-audio groove tracer. Our goal in this paper is to process the sonorine photos taken in Firestone Library and fit the pre-existing tools to the needs of this specific approach to retrieve measurable results.

We begin by first analyzing the images of the sonorines. Each sonorine has four photos taken of it, each with a different light angle. The 10319 by 7741 photos are of TIFF format in the RGB colour space at 16 bits per colour channel. Traditional PNG and JPEG image formats can only support 8 bits per channel, meaning that each of the R, G, and B values can range from 0 to $2^8 - 1$. TIFF formats are able to support 16-bit images, so the values in each colour channel can range from 0 to $2^{16} - 1$, allowing for higher granularity in colour details. Throughout this paper, all of our work we use to obtain results is done on full sized, 16-bit TIFF photos. In addition to photos of sonorines, we also analyze photos of blank paper and mirrored spheres, which were also taken from the same configuration with the same angles of lighting.

The photos of the blank paper are used to calibrate the sonorine photos, whereas the mirror sphere photos are used to determine the direction of light in the image. The GUI mentioned in the previous section of this paper was built for computing the light direction given a photo of mirror spheres, but it inaccurately estimates the shadow location and often crashes without warning to the point of unusable. This paper presents an alternate way of computing light direction using the GUI’s coordinate detection capabilities while avoiding its problematic operations.
Figure 1: A complete set of photos for one sonorine.

Figure 2: A complete set of blank paper photos.

Figure 3: A complete set of mirror sphere photos.

Once we obtain the light directions and process the sonorine photos, we can then use photometric stereo to create a normal map of sonorine’s surface, which can in turn be used to generate a height map and therefore sound. Figure 4 summarizes the steps necessary for audio recovery.

Figure 4: Order of operations from photos to sound.
4 Implementation

This project was implemented entirely in Python 3, with the extensive use of NumPy and OpenCV libraries, among others. While it is possible, and sometimes standard practice in other programming languages, to process images by iterating over the image pixel by pixel and modifying the individual pixels’ data, Python’s time-expensive loops makes this operation unacceptably slow. For example, early on in the project, we processed a 2582 by 1940 8-bit PNG image of a sonorine on a pixel-per-pixel basis and the runtime was slightly over 7.60 seconds.

There are a few alternatives to this, a couple of which we considered heavily: use the Python Imaging Library (PIL) or model the images as NumPy arrays and use NumPy operations. Both are significantly faster than pure Python operations as they are both implemented in C. PIL is quite robust and has a wide suite of image processing functionalities, including point operations, filtering with built-in convolution kernels, colour space conversions, resizing, rotation, and more. However, NumPy is commonly used with OpenCV’s Python binding, which provides all of the functionalities PIL offers plus robust computer vision tools. The developer community around NumPy and OpenCV are also more established than that of PIL. These factors, along with the fact that NumPy and OpenCV were also used in the project’s previous work, led us to settle on NumPy and OpenCV.

All grayscale and colour images in this paper are represented as 2D and 3D NumPy arrays, respectively, with two of the dimensions being the height and width of the image. The colour channels (only applicable to colour images) make up the third dimension. Using this representation, the same processing operation on the same 2582 by 1940 PNG takes around 0.59 seconds, a significant improvement over the pixel-by-pixel operation.

4.1 Photometric Stereo Surface Normal Map Computation

Robert J. Woodham first introduced photometric stereo in 1980 as a novel technique to analyze multiple images of an object under varying lighting conditions to determine a normal vector at each pixel. The surface normal map can be determined under his original assump-
tions (Lambertian reflectance\(^1\), distant point light sources, uniform albedo\(^2\)) by inverting the following linear equation:

\[ I = L \cdot n \] (1)

Where \(I\) is the matrix of \(m\) observed intensities, \(L\) is a matrix of normalized light directions, and \(n\) is the unknown surface normal. The Moore-Penrose pseudoinverse offers a generalization of the inverse for 3 or more lights by multiplying both sides of the equation by \(L^T\), giving us

\[ L^T I = L^T (L \cdot n) \]

\[ n = (L^T L)^{-1} L^T I \] (2)

In this paper, we have \(L\) as a matrix with the four light directions from which the sonorines were photographed and \(I\) as a matrix with the 4 processed sonorine images showing the observed intensity at each pixel.

### 4.2 Calculating Light Direction in Photo

We know that \(L\) requires the direction of the point light sources that is casting light on the sonorine. That is, we need a normalized vector in 3D space that points from the sonorine to the light source for every angle of light on the sonorine. To do this, we analyze the highlights on the mirror spheres in the images taken in Firestone library and make some assumptions that will help simplify our calculations without sacrificing too much accuracy. The assumptions are as follows:

- The camera is directly above every mirror sphere and is infinitely far away.
- The light source is a point infinitely far away.
- The variance of light across the sonorine is negligible; that is, one light vector can be chosen to represent all light vectors across the sonorine to a high degree of accuracy.

Although the GUI mentioned in 2.2 cannot reliably be used to compute light direction, it is able to take a photo as input and provide the following key pieces of information on each mirror sphere, which can then be extracted and used in our own calculations:

\(^1\)The property that defines an “ideal” diffusely reflecting surface, such as freshly fallen snow or white paper. Since phonocards have a similar surface to the ideal surfaces, Woodham’s assumption can be held without introducing much error.

\(^2\)The albedo is defined as the ratio between the reflected light and total incident light over a unit area. It is expressed as a constant between 0 and 1, where 0 is total light absorbance and 1 is total reflectance.
• $x$ and $y$-coordinates of the center of sphere.
• $x$ and $y$-coordinates of the center of highlight on the sphere.
• radius of the sphere.

The following diagram illustrates how we can use the above information to compute the vector to the light.

![Diagram of a camera view of a mirror sphere (xy-plane).](image)

Figure 5: A camera view of a mirror sphere ($xy$-plane).

We first look at the sphere from the perspective of the $xy$-plane. We can simply find the Euclidean distance $d$ from the center of the sphere to the center of the highlight using:

$$d = \sqrt{dx^2 + dy^2}$$  \hspace{1cm} (3)

where $dx = hx - cx$ and $dy = hy - cy$. $dx$ and $dy$ represent the $x$ and $y$-components of the light vector we are calculating. To find the remaining component $dz$, we can investigate the cross-section of a mirror sphere in the plane of the direction of $d$. Denoting the angle of reflection (angle between the light vector $l$ and the sphere’s normal vector $n$) as $\theta$, a sphere’s cross-section can be shown in Figures 6 and 7.

With the above in mind, we can compute $dz$ as follows:

$$\theta = \arcsin \left( \frac{d}{r} \right), \quad \phi = \left( \frac{\pi}{2} \right) - 2\theta$$

$$dz = d \tan (\phi)$$

$$dz = d \tan \left( \frac{\pi}{2} - 2 \arcsin \left( \frac{d}{r} \right) \right)$$  \hspace{1cm} (4)

Therefore, for each mirror sphere, we have our light vector $l$ that points from the sphere’s surface to the point light source as

$$l = \text{norm}([dx, dy, dz])$$  \hspace{1cm} (5)
where

\[ dx = hx - cx, \quad dy = hy - cy, \quad dz = d \tan \left( \frac{\pi}{2} - 2 \arcsin \left( \frac{d}{r} \right) \right) \]  

(6)

### 4.2.1 Selecting the Medoid

We now have one light vector for each sphere in each mirror sphere photo, for a total of 16 vectors. However, we can only use one vector per photo in our normal map computations, as per photometric stereo. We can simply use the mean of the vectors to represent the photo, but since even small errors in sphere coordinate and highlight detection can result in noticeable differences in light vector calculation, an outlier or two can sway the mean undesirably. To avoid this, we select the medoid of the vectors. Commonly used as a representative object in data clustering, the medoid expresses the data point that is closest to all other data points. It is typically only used on two-dimensional data, so we derived a custom definition to be used with 3D vectors. Let \( l_1, l_2, \ldots, l_n \) be a set of \( n \) vectors in 3D space. Our medoid is then defined as

\[ l_{\text{medoid}} = \text{argmin}_{x \in \{l_1, l_2, \ldots, l_n\}} \sum_{i=1}^{n} x \cdot l_i \]  

(7)

Four medoid light vectors from the four mirror ball images can now be used as part of the surface map computation.

### 4.3 Image Processing

Obtaining the matrix of light vectors is only one of the two required parts of our surface normal map computation. The other is obtaining the matrix of observed pixel intensities for every direction of light from which the sonorine was photographed.
4.3.1 Tackling Light Variance Over the Sonorine

We use photometric stereo to observe the change in behaviour of a pixel as it is illuminated from different directions, but the illumination itself poses a challenge as it is casting uneven levels of light throughout the photographed surface. Because of this, a pixel at a particular location may be darker in one photo than another. Although this may seem like reasonable practice when observing a pixel under varying light conditions, it is in fact undesirable because we are interested in the change in pixel behaviour due to changes in light direction and not other factors that may affect intensity, such as distance from the surface. After all, one of the basic assumptions of photometric stereo (and also one of our assumptions for light vector calculations) is that the light is coming from a point light source infinitely far away. We want to isolate the variable for change in intensity to be nothing other than change in light direction.

We can do this by normalizing the image of the sonorine with a corresponding image of a blank piece of paper taken with light of the same direction. That is, we can divide the pixel colour values of the sonorine image with that of the blank paper to eliminate any discrepancies in intensity due to light distance. Since we represent our images using NumPy arrays, dividing corresponding pixels in two images is a simple and quick operation. Below is an outline of our normalization algorithm:

1. For each of the 4 sonorine-paper image pairs, divide the sonorine image by the paper image to normalize it.
2. Find the maximum value $m$ across all 4 normalized images and divide all 4 images by $m$. It may be possible that a pixel’s colour values in the sonorine image is larger than that of the paper image and the result is greater than 1, when the normalized colour values can only range from 0–1. This step scales all the images to ensure no value is greater than 1.
3. Scale pixel colour values of the images to $[0, 2^8 - 1]$ and save them as 8-bit PNGs.

We experimented with this algorithm on several resized PNGs of sonorines. These PNGs were 8-bit, in the RGB colour space, and were 2582 by 1940 in size. One of the results is shown in Figure 8.

4.3.2 Improving the Normalization Algorithm

The normalization was quite successful on the resized PNGs, but we identified and implemented a few areas of improvement for operation on full-sized TIFF images.

We were dividing across all 3 colour channels (red, green, blue) when working with PNGs, but we are only looking for the images’ observed intensities and we can retrieve that
information from grayscale images. Since grayscale images only use one colour channel, we can save significant amounts of computation and memory by performing all processing operations on grayscale images. OpenCV has built-in functions for changing colour spaces of images, allowing us to easily convert our images to grayscale before normalizing. We also applied OpenCV’s built-in median blur to the blank paper images to smoothen out any blemishes that may cause outlier values during division.

The images our algorithm is to run on are 10319 by 7741 16-bit TIFF photos from the Firestone Library camera. Such images are not only large in dimension but also fine in detail (each image is slightly upwards of 480 MB in size!). We want to reduce unnecessary computation and runtime, so we cropped the sonorine images to only the portion containing the sonorine and the same section of the image from the corresponding blank paper photo. The resulting images were at a more reasonable 4800 by 4800 in dimension and 140-150 MB in size. The results from our revised algorithm on the cropped TIFF images can be seen in Figure [9].

We examined the resulting images carefully and identified an issue relating to the scaling of the minimum and maximum pixel values to an image’s acceptable range (0 to $2^8 - 1$ for PNGs and 0 to $2^{16} - 1$ for TIFFs). A few occasional pixels are much brighter than others due to uneven exposure to light, bumps and scratches on the sonorine, image noise, and other factors. Setting them to be the maximum intensity would cause the average pixel to be scaled much lower, producing a relatively dark image with some occasional bright white regions as shown in Figure [10].

Our solution to this is to clip the pixel values at the 4th and 96th percentiles. That is, we set the minimum and maximum values of the image at the 4th and 96th percentiles, respectively, and set all values lower than the minimum to the minimum as well as all values higher than the maximum to the maximum. This is to ensure that the outlier pixels are not skewing the overall shading of the image. We did not want the percentile range to be too narrow, as that would over-unify the observed intensities needed in photometric stereo.
We experimented with different percentiles cut-offs and decided that the 4th and 96th was the widest range that gave us a clear image of the sonorine grooves against a rather uniform white backdrop. The result is shown in Figure 11.

In summary, our normalization process is as follows:

1. Crop images to pixel coordinates that would contain the sonorine.
2. Convert all images to grayscale and apply a median blur to the paper images.
3. For each of the 4 sonorine-paper image pairs, divide the sonorine image by the paper
4. Find the 4th and 96th percentile values across all 4 normalized images and clip all 4 images to that range. The range is sampled across all 4 images to ensure consistency between them.

5. Scale pixel colour values of the images to $[0, 2^{16} - 1]$ and save as 16-bit TIFFs.

4.4 Photometric Stereo Verification

After obtaining our normalized images of observed intensities, we can visualize the application of photometric stereo. This visualization is important because it allows us to verify that photometric stereo is in fact effective on these images. The only other visual checkpoint after this step is the final height map itself, so verifying the effectiveness of photometric stereo allows us to catch any issues with the computations before they make their way into our final results.

The general goal is to artificially shine different coloured lights on the sonorine so we can observe the change in colour as the pixels are lit from different directions. Let us denote the four normalized sonorine images by the corner at which the light in the image originated: NW, NE, SW, and SE. We combine the images by subtracting images with opposite light directions: NW – SE and NE – SW. Let the result of the first subtraction be $sub_0$ and that of the second subtraction be $sub_1$. From there, we then experimented with two different approaches to colour the visualisations.
4.4.1 RGB Method

The RGB method is used to illuminate the sonorine with blue light on one side and red light on another, while keeping a constant green throughout the image. The implementation is the following:

1. Create a new image in the RGB colour space.
2. Scale $sub_0$ to $[0, 2^8 - 1]$ and assign to the red colour channel. This visualization is for verification and does not require the fine colouring details of a 16-bit TIFF, so we work in 8-bit PNGs to save computing time and memory.
3. Scale $sub_1$ to $[0, 2^8 - 1]$ and assign to the blue colour channel.
4. Assign a constant value of 100 to the green colour channel.
5. Save the new image.

The results are as follows

![RGB visualization](image)

Figure 12: RGB visualization of the entire sonorine (left) and crops (middle and right).

4.4.2 HSV Method

The HSV (Hue, Saturation, Value) method is the more sophisticated and experimental approach. We used this method to experiment with incorporating a broader range of fluidly transitioning colours rather than being limited to the 3 colours of the RGB model. The implementation is as follows:

1. Create a new image in the HSV colour space.
2. For each value \((x, y)\) in \((\text{sub}_0, \text{sub}_1)\), compute the angle (in degrees) between the positive \(x\)-axis to a vector extending to \((x, y)\) from the origin to the positive \(x\)-axis. Scale the values to 0-179 and assign to the H channel.

3. For each value \((x, y)\) in \((\text{sub}_0, \text{sub}_1)\), find the magnitude of the vector extending to \((x, y)\) from the origin and scale it to \([0, 2^8 - 1]\). Assign these values to the S channel.

4. Set the V channel to a constant equal to \(0.5 \cdot 2^8 - 1\).

5. Save the new image.

![Figure 13: The entire sonorine (leftmost) and crops from HSV visualization.](image)

While the HSV visualizations have greater contrast compared to RGB ones, they are slightly too removed from realistic images of sonorines (e.g. the shadows are bright yellow instead of the RGB method’s dark green). The RGB method was adept in both displaying colour variance and preserving realism in the image, so we considered it to be the more successful visualization. Either way, these visualizations show that photometric stereo can effectively be used with the sonorine images.

5 Evaluation

We evaluate our results in three ways, corresponding to our sound recovery process referenced in Figure 4. We first evaluate the recovered normal map, then the height map, and finally the audio.

5.1 Normal Map Results

We were able to successfully compute the normal map using a function from previous mathematical work with very little modification. To visualize the sonorine’s map of normal vectors,
we can generate artificial images using the dot product of normal vectors with the light direction vectors we found earlier. If the artificial images resemble the actual images of sonorines, we know that the normal maps are accurate.

For each pixel, with normal vector \( n \) and light vector \( l \), we compute

\[
max(0, n \cdot l)
\]

and scale the value between 0 and \( 2^8 - 1 \). Writing the values to an image would give us a black and white visualization of the normals relative to the direction of light, where the whitest pixels represent normals that are close to parallel with the light vector and dark pixels represent ones that are close to perpendicular. The resulting images are as follows:

![Figure 14: The 4 artificial images generated from the normal map and 4 light vectors.](image)

By visual inspection, we can confirm that the artificial images closely resemble the original sonorine images, both in shape and light reflectance. The closer viewings of the artificial images show that they outline the same path and contain the same textures as the normalized photos from Section 4.3.2.

We can also see from our RGB visualization (Figure 12) the effect of light direction on the sonorine’s grooves. The same effect can be seen from our artificial images: the white regions are where the light strikes the surface most parallel to its normal. The regions lit most brightly by blue and red light in the RGB visualization are also the brightest in the artificial images, with respect to their light direction.

Because our normal map is still able to generate a trustworthy image of the sonorine with the light vector used to compute it, we can conclude that the normal map is accurate.

### 5.2 Height Map Results

We use our normal map computed earlier to retrieve the height map. We were able to successfully retrieve a height map (a value in \( z \)-direction for every pixel \((x, y)\)) and plotted the height values as a heatmap. The height map solver also allowed us to define the number of
Figure 15: Close crops of artificial images (left) compared to crops of normalized images from 4.3.2.

iterations, which corresponds to the iteration limiter on the least squares regression function used to estimate a smooth surface. The higher the number of iterations, the smoother the height map will be. We experimented with 50, 100, and 150 iterations.

Figure 16 shows the height map of a full sonorine. Figure 17 are cropped from the height maps to show more detail and directly compare the results of a varying number of iterations.

We notice that some of the grooves only faintly continue throughout some areas of the sonorine, and may even fade away due to excessive smoothening at higher iterations. Unfortunately, this is not a desirable effect as it may cause the sound reproduction program, which mimics the path of a needle through the grooves, to not pick up the necessary details or even cause the path to derail. We thought this effect may be due to the whitening of the sonorine’s background when clipping the normalized images to the 4th to 96th percentile intensity range, but the height map remained relatively unchanged even with normalizations at the full intensity range.

As a comparison, we visualized a height map computed from a flatbed scan of the same sonorine.

We can see that the fading of the grooves is more prevalent in our height map, and that the grooves in the flatbed scan are more uniform in height. It is unclear, without inspecting the sonorine in person, whether or not our approach reveals more information about the depth of the groove (and as a result may be able to capture a wider array of audio), or if it
Figure 16: Height map produced by 150 iterations.

Figure 17: Height maps produced by 50 (left), 100 (center), and 150 (right) iterations.

is just more error-prone.

It is interesting to note that the grooves in the flatbed scan’s height map are a light yellow, indicating that they are more elevated than the rest of the sonorine. In our height map, the grooves are a darker blue, which represents impressions in the card stock, but it appears to be the opposite with the height maps from the flatbed scanner.
5.3 Sound Results

When we ran the sound recovery groove tracer on our height map, the correct starting location failed to be detected and the tool started tracing the groove path near the outside edge of the sonorine. As a result, we were only able to obtain a small snippet (around two seconds) of audio. The audio was quite noisy and too short to make out anything distinguishable. The issue persisted on multiple attempts and sonorines, as well as through our debugging attempts.

The green marker in the Figure 19 shows the starting point of the path, while the dark blue lines show the path itself and the red dots are areas of uncertainty in the groove.

In the process of generating audio, we plotted some data to get a better understanding of our situation and to quantitatively and visually compare our audio results with the results from the flatbed scanner. The comparison was performed on the same sonorine.
Figures 20 and 21 show the radius to the center of the sonorine with respect to the radians traversed in the traced path. Both graphs show the correct positive relationship. Our graph displays higher variance visually, but is on a smaller scale in both the horizontal and vertical axes. The general upward trend in both graphs indicate that that the paths are traced properly and there is no major derailing of the tracer.

Figures 22 and 23 show the radius to the center of the sonorine with respect to the radians traversed in the traced path. Both graphs show the correct positive relationship. Our graph displays higher variance visually, but is on a smaller scale in both the horizontal and vertical axes. The general upward trend in both graphs indicate that that the paths are traced properly and there is no major derailing of the tracer.

We also used the change in groove height and sample number to represent frequencies in the recovered audio, up to the 10 000th sample. Since the path starting location was not the same, the first 10 000 samples corresponded to two different sections of the sonorine for the two audio files. However, we can see a general pattern that the variation in height across samples are similar, with our height map having a slightly larger negative range. The noise from our height map is noticeably higher, possibly due to the lack of clarity in our grooves.
in comparison to the very uniform grooves from the flatbed scanner.

6 Discussion and Future Work

6.1 Discussion of Results

The goal of this paper is to introduce a novel technique to recover audio from photographs of sonorines. We first analyzed high fidelity photos of sonorines, blank paper, and mirror spheres to deduce normalized light direction vectors that point from the surface to the light source. We processed the sonorine photos by normalizing them with photos of blank paper to obtain an image of observed intensities. We used our light direction vectors and intensity images to generate a normal map of the sonorine from photometric stereo, from which we then generated its height map. Finally, we used the height map data to reproduce the sonorine’s audio. While we were able to recover some audio, we still recommend the previous approach of using flatbed scanners to image the sonorines.

The major difference between our photovisual method and the flatbed scanner one can be seen in the clarity of grooves in the height maps. The height maps generated from flatbed scans contained grooves that appeared solid and consistent in form, while still revealing much of the texture within the groove. The grooves from our photometric stereo approach were not as clear and occasionally faded out, leading to more inconsistency in the audio output both in terms of audio quality and amount of audio retrieved.

There are many potential reasons for this difference, perhaps the largest of which is inaccuracy in the light directions computed in Section 4.2. The flatbed scanner casts light uniformly across the sonorine, while we try to mimic the effect from point light sources. To do this, we made several assumptions outlined in Section 4.2, but those assumptions may not be wholly accurate. For example, we assumed that the variance of light across the sonorine is negligible. Since traditional photometric stereo takes one light vector for every intensity matrix as input, we represented light in the entire image with the medoid of 16 light vectors.
This representation may be oversimplified and caused some loss of detail during our height map computation. Our other assumptions of the camera being directly above every mirror sphere and the light source being infinitely far away may have also contributed slightly to inconsistencies in the height map.

Another likely reason is the approach to height map computation. Throughout this paper, we treated the mathematical techniques for height map generation and the groove tracer tool as more or less blackboxes, modifying minor details for our use but keeping most of the functionality untouched. Since past work was completed using flatbed scans, we formatted our data in accordance to previous input requirements to use the same tools. However, it is likely that the approach to computing the height map from our method varies slightly from that of the flatbed scans. A detailed understanding and modification of the techniques behind height map generation is out of the scope of this paper, but may be necessary to accurately compute height maps for our purposes.

The phenomenon of grooves appearing more elevated than the rest of the surface in the flatbed scans’ height map is unexpected and still requires further investigation.

### 6.2 Future Work

To address the issues mentioned above, a reasonable first step to improve on this paper would be to develop a more sophisticated method to account for the variance in light direction across the sonorine. This would result in multiple light vectors corresponding to one intensity matrix, and since traditional photometric stereo only accepts one light vector, developing a custom photometric stereo approach would be required. Doing so would remove a couple large assumptions from our light vector computation process and can hopefully result more robust height maps.

We can also take some time to holistically comprehend the approach and code in the height map computation tool as well as the groove tracer for audio recovery. Doing so would allow us to properly identify any differences in the height map generation process and modify the approach accordingly to tailor to our method. Furthermore, a better understanding of how a path’s starting point is detected in the sound recovery program may be useful for troubleshooting when program fails to detect the proper starting location. We collaborated with the developers of the aforementioned tools throughout this paper, but further collaboration is required to fully understand the tools’ inner workings.
7 Acknowledgements

First and foremost, I would like to thank my advisor Adam Finkelstein for his invaluable guidance and advice throughout this project. Additionally, my work also involved the work of several others, and I would like to give special thanks to Mariah Crawford ’22 for explaining to me and providing me with code for the light detection GUI, Rohit Narayanan ’22 for meeting with me early on in the semester to go over math involved with height map recovery and continued collaboration throughout the semester, and Ezra Edelman ’23 for guiding me on using the groove tracing sound recovery tool he developed over two summers.
References

[1] “The phonocard,” *Scientific American*, vol. 93, no. 13, p. 235, 1905.

[2] P. Cavallari, “Le phonopostale et les sonorines: un échec riche d’idées,” pp. 77–86, Dec. 2016.

[3] C. Zandonella, “Dean for research innovation funds awarded to highly exploratory projects.” [https://www.princeton.edu/news/2017/03/22/dean-research-innovation-funds-awarded-highly-exploratory-projects?section=topstories](https://www.princeton.edu/news/2017/03/22/dean-research-innovation-funds-awarded-highly-exploratory-projects?section=topstories), Accessed: 2019-09-16.

[4] W. Clarkson, T. Weyrich, A. Finkelstein, N. Heninger, J. A. Halderman, and E. W. Felten, “Fingerprinting blank paper using commodity scanners,” in *Proc. IEEE Symposium on Security and Privacy*, May 2009.

[5] R. Woodham, “Photometric method for determining surface orientation from multiple images,” *Optical Engineering*, vol. 19, Jan. 1992.

[6] M. Varma and A. Zisserman, “Estimating illumination direction from textured images,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, June 2004.

[7] V. Fadeyev and C. Haber, “Reconstruction of mechanically recorded sound by image processing,” *Journal of the Audio Engineering Society*, vol. 51, Mar. 2003.

[8] Northeast Document Conservation Center, “Audio preservation with irene.” [https://www.nedcc.org/audio-preservation/irene](https://www.nedcc.org/audio-preservation/irene), Accessed: 2019-11-30.

[9] O. Johnsen, S. Stotzer, F. Bapst, and R. Ingold, “Detection of the groove position in phonographic images,” in *2007 IEEE International Conference on Image Processing*, vol. 6, Sept. 2007.

[10] Swiss National Sound Archives, “Visualaudio.” [https://www.fonoteca.ch/gallery/visualAudio/home_en.htm](https://www.fonoteca.ch/gallery/visualAudio/home_en.htm), Accessed: 2019-11-30.

[11] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma, “Robust photometric stereo via low-rank matrix completion and recovery,” in *Proceedings of the 10th Asian Conference on Computer Vision - Volume Part III*, ACCV’10, (Berlin, Heidelberg), p. 703–717, Springer-Verlag, 2010.