A MATLAB Program for Quench Location in ILC SRF Cavities Using OST Detectors of Second Sound

Edward B. Lochocki
Physics and Astronomy Department, Rutgers University, Piscataway, NJ, 08854
(Dated: August 6, 2009)

The large number of currently proposed particle accelerators, including the International Linear Collider (ILC) and Cornell's own Energy Recovery Linac (ERL), will require unprecedented quantities of high-quality superconducting RF (SRF) cavities. Quench detection techniques must therefore be quick, cheap, and reliable so that defects can be located and fixed with a minimum of time and effort. Oscillating superleak transducers (OSTs) detect the second-sound waves emitted by quenches and can be used to triangulate to the quench location. OSTs are inexpensive, do not require extra cold tests to use, and have a high signal-to-noise ratio; thus, they offer the best available method for quench detection. Our newly developed MATLAB program, SecondSound 2.0 (SS2), enhances their ease-of-use even further.

I. INTRODUCTION

Quenching occurs in superconducting cavities when small surface defects (pits, bumps, particles, and/or grain boundaries) create abnormally large electromagnetic fields in the regions surrounding them. The high RF losses generated in these areas produce heat and raise the local temperature above the cavity's critical temperature, thereby forming normal-conducting spots. The cavity then rapidly becomes thermally unstable and the normal-conducting region grows until all stored energy has been dissipated.

Due to the thermal nature of quenches, all quench location methods focus on detecting their heat signatures. A popular method involves using an array of thermometers on the surface of the cavity to find the region of highest temperature. However, this method has a major drawback in that it requires two or three cold tests to locate the quench. One test is needed to determine that the cavity quenches, and up to two more tests are needed to actually locate the quench with the thermometer arrays [1]. Each test can take several hours and consume large amounts of liquid helium, and so a method that can find quench spots using fewer tests would be more desirable.

A more clever method involves utilizing an interesting property of the superfluid helium (He-II) coolant itself. Along with familiar pressure/density first-sound waves, superfluid helium also exhibits temperature/entropy waves known as second-sound [3, 5]. So when a cavity quenches, second-sound waves are emitted from the quench spot. (Quenches actually dump so much heat into the helium that the wave exceeds the Landau critical velocity and travels faster than second-sound until it later couples to a second-sound wave [4].) The clever idea is that several temperature-sensitive resistors can be placed around the cavity, and if it quenches, they can be used to determine the arrival times of the second-sound waves. Since second-sound velocity in liquid helium is a well-measured functions of temperature, travel distances can be computed, and if three or more resistors measure a signal, the quench location can be found by triangulation. Although this technique can locate a quench in a single cold test, its main disadvantage is that temperature-sensitive resistors have low signal-to-noise ratios because they do not exclusively measure second-sound waves and their resistances change very little with temperature.

II. THE OSCILLATING SUPERLEAK TRANSDUCER METHOD FOR QUENCH DETECTION

OSTs have been described extensively in the literature [6, 7, 12] and used in a variety of ways. For our purposes, we will consider them to be second-sound microphones, although they can also be used to generate second-sound waves. As a brief overview, each OST consists of three main components: a superleak,[1] an aluminum casing, and a brass center. In an OST, the superleak is in electrical contact with a grounded aluminum casing that holds down its edges. Inside of and insulated from the aluminum casing is a brass center at 140 V. The superleak rests on top of this, forming a capacitor.

When a second-sound wave reaches the OST, the superfluid component of the helium rushes in and out, causing the superleak to first bellow out and then to return to its original position. This creates an easily-measured change (of several orders of magnitude) in the capacitance of the OST. Thus, OSTs can be used to find quench locations in the same way as temperature-sensitive resistors, but with greater accuracy, due to their much higher signal-to-noise ratios.

Two techniques previously existed for locating quench spots using OST data:

1. The "Wire Method" was the first technique for locating a quench using OST data and the only one to successfully be used. First, wave-travel times were found by subtracting the time at which the cavity quenched from the time at which the second-sound wave arrived at each OST. Since second-sound velocity could easily be found from the temperature of the test, the distance from each OST to the quench could be computed. Wires were cut to the correct length and attached to each OST, and their free ends were moved around until they all (approximately) met in one spot. The advantage of this method was that the positions of the OSTs and the geometry of the cavity did not need to be measured or recorded as long as everything was undisturbed after the test. However, once the OST/cavity assembly was taken apart, the analysis could never be redone. An even more serious downside was that this technique disregarded the fact that quench signals travel faster than second-sound before coupling to second-sound waves.

2. SecondSound 1.0 (SS1) was the first attempt at making a computer program that could determine quench locations from OST data. A computer-based quench location system has an advantage over the manual "Wire Method" in that all information about the test is stored in files that can be loaded later so that the setup and results can be revisited; however, a downside is that OST positions and cavity geometry parameters must be known precisely. SS1 created an analytic profile function for a standard TESLA cavity with user-input ellipse parameters, stored the location, size, and orientation of the OSTs, read in the OST data from an oscilloscope data file, and

[1] A superleak is a thin porous layer of cellulite covered in an even thinner layer of aluminum. Its pores are too small for the viscous normal-fluid component of liquid helium to pass through, but they allow the superfluid component to move freely. The aluminum layer on top allows the superleak to conduct electricity.

determined the location of the quench. Unfortunately, SS1 had some serious problems; it could only work with standard TESLA cavity shapes, it computed wave-travel paths incorrectly, it treated the surface as having no thickness, and like the "Wire Method," it disregarded the fact that quench signals travel faster than second-sound before coupling to second-sound waves.

III. SECONDSOUND 2.0

SS2 greatly improves upon SS1. The program now has built-in capability to generate standard TESLA cavities or re-entrant cavities, and it can import other cavity geometries from ASCII files. It uses a more advanced method to correctly determine wave-travel paths, it models the surface as having a thickness, and it takes into account the fact that quench signals travel faster than second-sound before coupling to second-sound waves. It demonstrates a practical and general method for computing quench locations and has the possibility to become the method-of-choice for locating quench spots with OSTs. The basic process at work in SS2 is as follows:

1. The temperature, the inner cavity surface, the cavity thickness, the oscilloscope data, and the locations, sizes, and positions of the OSTs are input (using a variety of methods). The temperature is used to find the velocity of the second-sound waves, and the oscilloscope data is used to find the time-of-flight of the waves on their way to each OST.

2. For each vertex V on the outer cavity surface, the wave-travel path between V and each OST is computed using a modification of the method described in [8].

3. For each cavity vertex/sensor pair, we now know the wave's total time-of-flight (Δt), the total distance traveled by the wave (Δd), and the second-sound velocity (v_{ss}). Thus we have two different distances: Δd and $\Delta t \times v_{\mathrm{ss}}$. Since the wave travels faster than second-sound for part of its path, Δd should be greater than $\Delta t \times v_{\mathrm{ss}}$ and we can find an error $\epsilon > 0$ using the formula

$$\epsilon = \Delta d - \Delta t \times v_{\mathrm{ss}} \tag{1}$$

4. For each vertex V, we now know ϵ for each OST. We can therefore find the standard deviation of the ϵ's for V. The cavity defect should be small enough to be considered a point source and the region of turbulence where the wave travels faster than second-sound should be a sphere, so we expect ϵ to be the same for each OST at the quench point. Therefore, the vertex closest to the quench location is the vertex that has the smallest ϵ standard deviation.

A. Inner Surface Definition

Most of the inputs (temperature, surface thickness, OST position, OST size, OST orientation, oscilloscope data) are simple and straightforward to input using either ASCII files or the SS2 GUI. While the cavity surface is also simple to input (using either an ASCII file or the built-in Model Creator), it requires some explanation first, as it has a few structural requirements it must satisfy. First of all, the surface is stored as a polyhedral approximation of the real cavity, rather than as a set of analytic functions. This makes the later calculations simpler and allows for exotic cavity geometries that cannot be described by simple functions. The set of surface points that make up the polyhedral approximation is stored in the form of three two-dimensional matrices: X, Y, and Z. For any point (i, j) in "surface space", the point $(X(i, j), Y(i, j), Z(i, j))$ is a point in real 3-D space that lies on the inner surface of the cavity. Although there are other ways to define polyhedra (for example, as lists of vertices and faces), this method is useful because its 2-D "surface space" structure can preserve information about the spatial relationship between points. We have a few requirements that ensure this fact (assuming that X, Y, and Z are $m \times n$ matrices):

1. The succession of spatial points corresponding to $(1, j)$, $(2, j)$, ..., (m, j) for any j must form a line along the beam axis of the cavity that lies in a single vertical plane.

2. The succession of spatial points corresponding to $(i, 1)$, $(i, 2)$, ..., (i, n), $(i, 1)$ for any i must form a ring around the cavity surface that lies in a single horizontal plane.

3. If $i \neq x$ or $j \neq y$, then the spatial point corresponding to (i, j) cannot be the same as the spatial point corresponding to (x, y).

Thus, for example, if we have a "surface space" point (i, j), we know that $(i + 1, j)$ and $(i, j + 1)$ are two of its nearest neighbors and that, together with $(i + 1, j + 1)$, the four points form a four-sided planar face of the polyhedral cavity approximation. This is extremely useful in later calculations. If the matrices are not defined in this way, the rest of the program will not work. See Figure 1 for a visualization of these requirements. To find the outer surface of the cavity, each point is shifted a distance t (where t is the cavity thickness) along the outward surface normal at that point. All later calculations are done using the outer surface of the cavity.

B. Distance Computation

Most of the calculations performed by SS2 were explained in Section III. However, the most complicated one was not. When a wave is emitted by the quench spot, it behaves as any wave would; it takes the shortest path to each OST that does not pass through the cavity itself. The determination of this path is a nontrivial operation. This problem, in its most general form, is called the 3-D Euclidean Shortest Path problem and has been plaguing computer scientists since 1984 when it was first proposed by [10]. It is possible to find the exact solution to the problem, but the algorithm runs in 2^{2^n} time, where n is the number vertices on the polyhedral obstacle [11]. The exponential time arises from the fact that the shortest path must travel through points on the edges of the obstacle; to find it, one must treat each edge as a continuum of points [9]. Papadimitriou developed an approximate solution to the problem in 1985; instead of treating each edge as a continuum of points, he picked only certain points on each edge to consider as nodes in the shortest path [8]. In our calculations, to decrease the time even further, we only consider the vertices themselves as possible nodes in the shortest path. We believe that errors caused by an approximate shortest path calculation will be overshadowed by errors resulting from OST position measurements and the fact that the polyhedral obstacle is an approximation of a cavity.

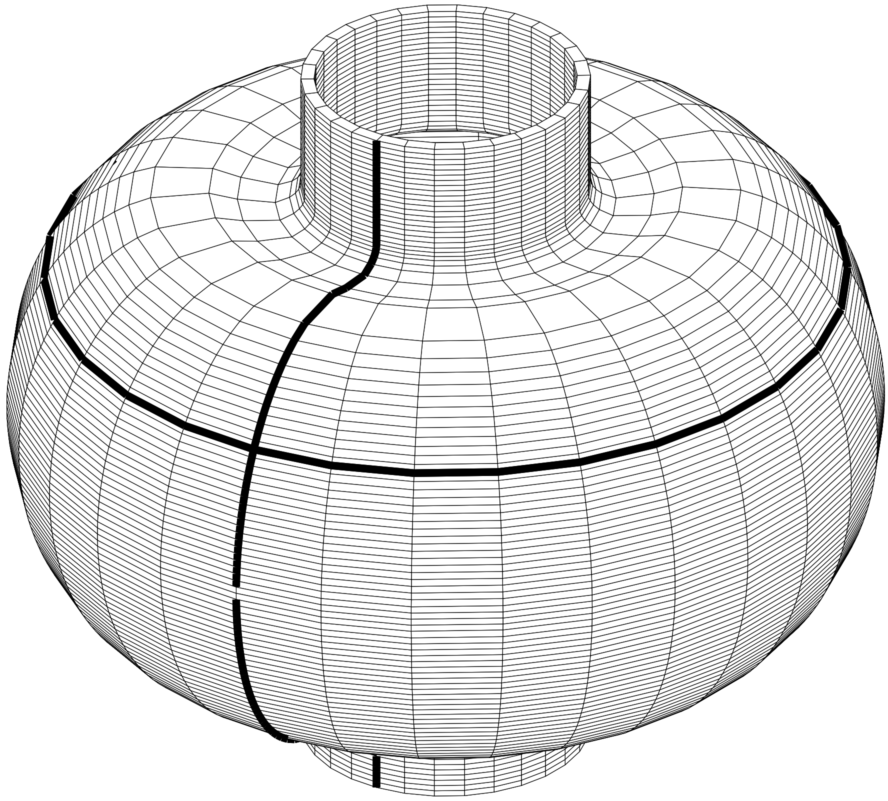

FIG. 1: An example surface made using the built-in Model Creator capability of SS2. The vertical line is a succession of "surface space" points (i, j) that all have the same j value. The horizontal ring is a succession of "surface space" points (i, j) that all have the same i value, and the last point is connected back to the first. Although these lines are shown on the outside surface of the cavity, there is a one-to-one correspondance between outer surface points and inner surface points and all the same relationships still hold.

To find the actual shortest path, we must first build a "visibility graph," which is essentially a set of rules about where the wave could go when it leaves a certain point. Structurally, the visibility graph is an $n \times n$ matrix, where n is the number of points in the scene; in our case, n is the number of OSTs plus the number of vertices on the outer cavity surface. Each entry is defined as follows: if vertex i has an unobstructed straight-line path to vertex j, then the $(i, j)^{\text{th}}$ entry of the visibility graph is the distance between the two vertices. Otherwise, if they cannot see each other, the $(i, j)^{\text{th}}$ entry of the visibility graph is infinity. The necessity of the requirements about the inner surface definition should now be clear, since they greatly speed up the visibility graph computation: if two vertices are on the same face, then we automatically know that they can see each other. Vertices may not be able to see each other for a variety of reasons; for example, if the line connecting them runs through the interior of the cavity or if it is blocked by a cavity face. Once the visibility graph is complete, we can find the wave-travel path between a cavity point and an OST by finding the sequence of points beginning at the cavity point and ending at the OST that yields the shortest distance. (To do this, we use a graph search algorithm called Dijkstra's algorithm

[2].) See Figure 2 for an example of a visibility graph and a shortest path.

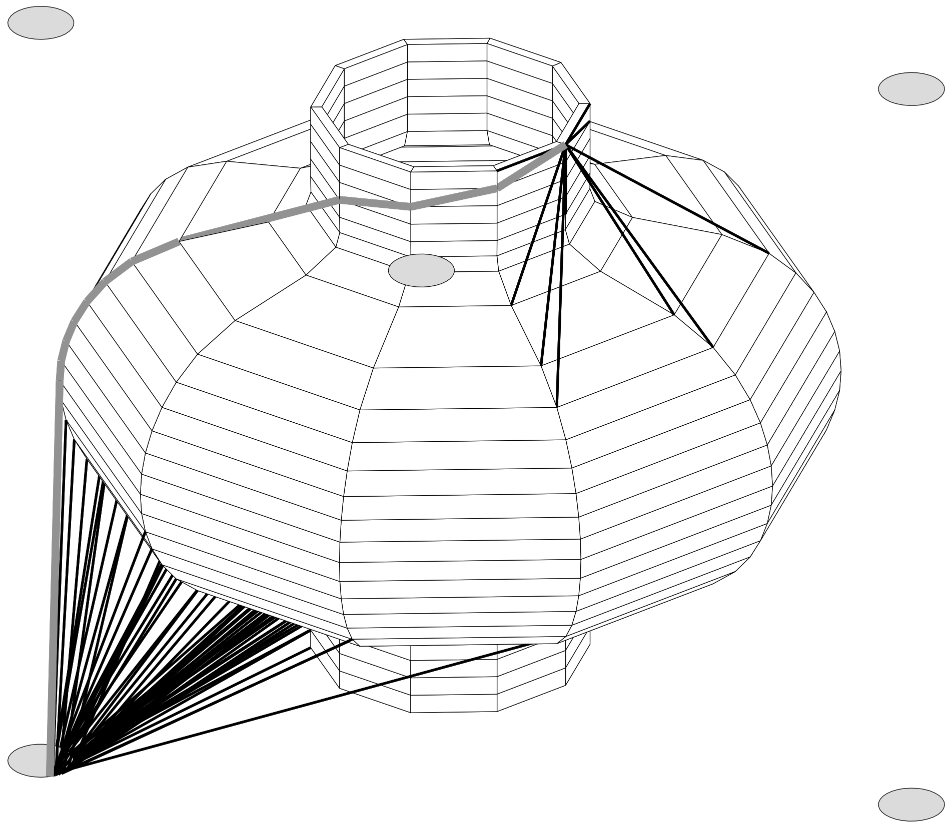

FIG. 2: An example of a visibility graph. Only two vertices are shown; one surface vertex and one OST. Heavy black lines are drawn to vertices that can see the two selected vertices. The thick gray line is the shortest path between the OST and the surface vertex. Note that lines connecting the OST to surface points go between the surface point and the closest point on the OST.

IV. RESULTS AND CONCLUSIONS

Figure 3 shows a sample result of the SS2 process (for physically meaningless data). All 8 OSTs were used in this analysis; however, SS2 allows the user to choose which OSTs to use. The error range is found by searching for vertices whose ϵ standard deviation (as defined in §III) is within a certain percentage of the minimum ϵ standard deviation. For this particular example, no additional points were within 10% of the minimum first-sound-time standard deviation, so the error range consists of a single vertex point (the quench point). Figure 3 was chosen to show a detailed view of the analysis; a screenshot of the actual MATLAB program is shown in Figure 4.

SS2 demonstrates a practical and useful method of computing quench locations with computers. Although it has a few downsides (the long visibility graph computation time and the chance for human error in OST position measurements), it also has some great benefits (a systematic approach to quench location, the ability to revisit a test after the OSTs have been removed from the cavity, and the capacity to work with many cavity shapes). Furthermore, it offers the possibility for direct interfaces to cavity-geometry programs such as Autodesk Inventor and to the OSTs via an ADC.

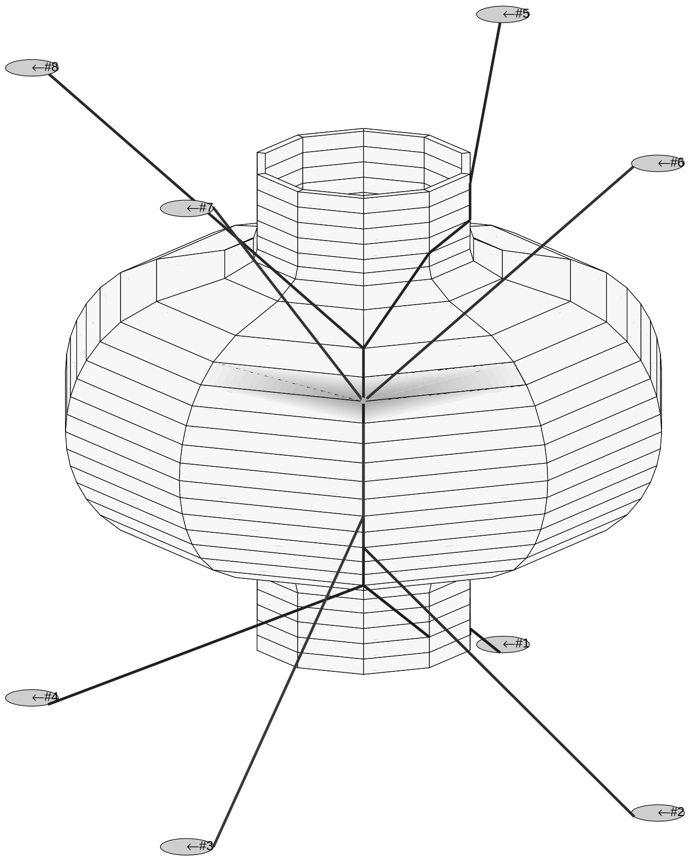

FIG. 3: Sample SS2 output.

V. ACKNOWLEDGMENTS

I would like to thank Professor Don Hartill for pioneering the use of OSTs in quench location and Zachary Conway and Professor Georg Hoffstaetter (all of Cornell University) for proposing this REU project and guiding me along the way. I must also acknowledge the many people at Cornell who helped make the REU possible: Rich Galik, Ernest Fontes, Lora Hine, Peggy Steenrod, and Monica Wesley. I am indebted to Justin Rodriguez, who wrote the code for SecondSound 1.0; I would still be trying to figure out MATLAB without his code as a guide. This work was supported by the National Science Foundation REU grant PHY-0849885.

---

[1] Conway, Z. A., Hartill, D. L., Padamsee, H. S., & Smith, E. N. 2008. Oscillating Superleak Transducers for Quench Detection in Superconducting ILC Cavities Cooled with He-II. *TTC*

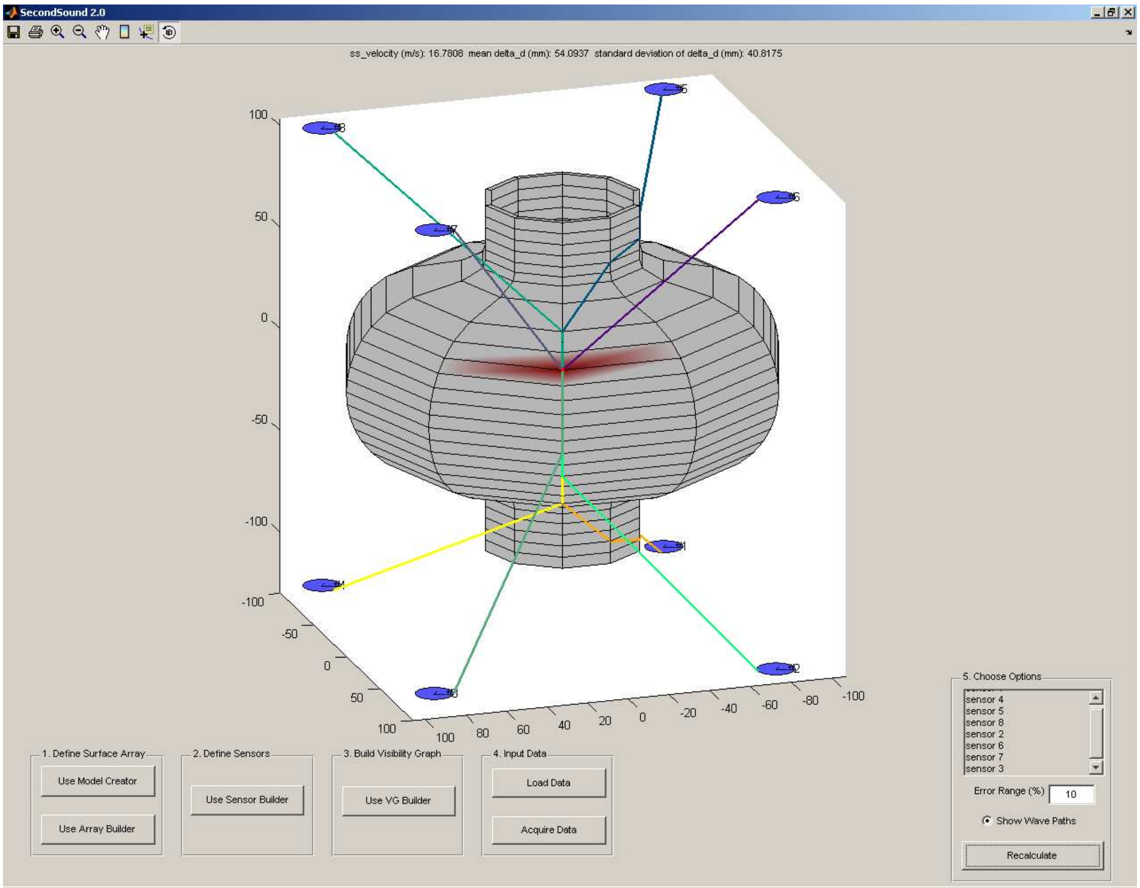

FIG. 4: Screenshot from SecondSound 2.0 showing the same results as Figure 3.

Reports.

[2] Dijkstra, E. W. 1959. A Note on Two Problems in Connexion With Graphs. *Numerische Mathematik*, **1**, 269–271.

[3] Kapitza, P. 1938. Viscosity of Liquid Helium below the λ-Point. *Nature (London)* , **141**, 74.

[4] Knauth, S., Hillmann, K., & Grill, W. 1999. Scanning Second Sound Microscopy with Vector Contrast. *Physica B: Condensed Matter*, **263-264**, 559–563.

[5] Landau, L. 1941. Theory of the Superfluidity of Helium II. *Phys. Rev.*, **60**(4), 356–358.

[6] Liu, M. 1984. Mechanical Detectors of Second Sound. *Phys. Rev. B*, **29**(5), 2833–2835.

[7] Liu, M., & Stern, M. R. 1982. Oscillating Superleak Transducer of Second Sound. *Phys. Rev. Lett.*, **48**(26), 1842–1845.

[8] Papadimitriou, C. H. 1985. An Algorithm for Shortest-Path Motion in Three Dimensions. *Information Processing Letters*, **20**(5), 259–263.

[9] Papadimitriou, C. H. 1986. Shortest-Path Motion. *Pages 144–153 of: FSTTCS*.

[10] Sharir, M., & Schorr, A. 1984. On Shortest Paths in Polyhedral Spaces. *Pages 144–153 of: STOC*.

[11] Sharir, M., & Schorr, A. 1986. On Shortest Paths in Polyhedral Spaces. *SICOMP*, **15**(1), 193–215.

[12] Sherlock, R. A., & Edwards, D. O. 1970. Oscillating Superleak Second Sound Transducers. *Review of Scientific Instruments*, **41**(11), 1603–1609.