



GEOMETRIC MICROPHONE ARRAY CALIBRATION BY MULTIDIMENSIONAL SCALING

Stanley T. Birchfield

Quindi Corporation, 480 S. California Ave., Palo Alto, California 94306
birchfield@quindi.com

ABSTRACT

Classical multidimensional scaling is a simple, linear technique for finding coordinates of points given their inter-point distances. In this paper we describe the algorithm and show how it can be used to solve the geometric microphone array calibration problem. The method requires no complicated hardware or calibration targets, just a tape measure (or similar measuring device). We also extend the basic algorithm to handle the case when some distances are unavailable, which makes the technique practical for microphone arrays with relatively large numbers of microphones.

1. INTRODUCTION

An important problem in microphone arrays is geometric calibration, viz., determining the locations of the microphones. These locations are prerequisite, for example, for acoustic localization and beamforming. Common solutions to the problem require expensive calibration targets and/or nonlinear optimization techniques that are subject to local minima [3, 4].

Multidimensional scaling (MDS) is a field of study concerned with embedding a set of points in a low-dimensional space so that the distances between the points resemble as closely as possible a given set of dissimilarities between objects that the points represent [1, 2]. For decades MDS has been a popular technique for analyzing experimental data in the physical, biological, and behavioral sciences.

Metric multidimensional scaling is a subset of MDS in which the dissimilarities are themselves (possibly noisy) distances. More precisely, the metric MDS problem is the following: Given noisy distances between a set of points in a Euclidean space, estimate the coordinates of those points. As an example, Figure 1 shows a map of the United States computed from the driving distances between major cities published in a road atlas, using the technique described in Section 2. Even though the driving distance between two cities is a poor approximation to their actual distance, the resulting map is accurate.

In this paper we show that metric MDS yields a simple, linear algorithm for determining the microphone locations

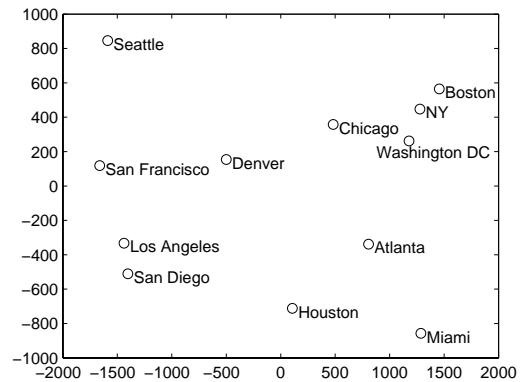


Fig. 1. Map of the U.S. based on driving distances between major cities, using classical multidimensional scaling.

given only their pairwise distances. Thus, with a simple tape measure (or similar measuring device) and a few lines of matrix algebra, the coordinates of all the microphone coordinates can be found. After presenting the basic algorithm in Section 2 below, we then extend it to handle the case when not all the distances are available. This extension makes the algorithm practical even for microphone arrays with a relatively large number of microphones.

2. CLASSICAL MULTIDIMENSIONAL SCALING

Suppose we have n microphones in a p -dimensional space (usually $p = 3$). Classical multidimensional scaling, which is a specific technique developed by [5] and made popular by [6] for solving the metric MDS problem, provides a simple procedure for finding their locations as follows [2].

First construct a squared-distance matrix D where each entry d_{ij} is the squared distance between microphones i and j : $d_{ij} = \delta_{ij}^2 = (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)$, where δ_{ij} is the distance between microphones i and j . From D compute the inner product matrix $B = -\frac{1}{2}JDJ$, where $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the double-centering matrix and $\mathbf{1}$ is a vector of all ones.

Without noise, $B = XX^T$, where $X = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T$



is the $n \times p$ matrix of coordinates, and hence $\text{rank}(B) = \text{rank}(XX^T) = \text{rank}(X) = p$. Since B is symmetric positive semi-definite, it may be decomposed as $B = V\Lambda V^T$, where $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, the diagonal matrix of eigenvalues of B , and $V = [\mathbf{v}_1, \dots, \mathbf{v}_n]$, the matrix of corresponding unit eigenvectors. For convenience the eigenvalues are labeled so that $\lambda_1 \geq \dots \geq \lambda_n \geq 0$. Since B is of rank p , it has p non-zero eigenvalues and $n - p$ zero eigenvalues and hence may be written as $B = V_p \Lambda_p V_p^T$, where $\Lambda_p = \text{diag}(\lambda_1, \dots, \lambda_p)$ and $V_p = [\mathbf{v}_1, \dots, \mathbf{v}_p]$. The coordinate matrix X is then given by $X = V_p \Lambda_p^{\frac{1}{2}}$.

With noise (i.e., the measured distances are imperfect), B will not be of rank p but rather will be full rank in practice (i.e., $\text{rank}(B) = n$). Conveniently, $X = V_p \Lambda_p^{\frac{1}{2}}$ still yields the optimal estimate of the coordinates, in the sense that $\sum_{i=1}^n \sum_{j=1}^n (\delta_{ij}^2 - \hat{\delta}_{ij}^2)^2$ is minimized, where $\hat{\delta}_{ij}$ is the estimated distance between the two microphones. In the terminology of principal components analysis (PCA), V_p contains the p eigenvectors that capture the most significant variation in the data.

By the procedure just described, the coordinates X of the microphones may be estimated from the distances between them. The resulting coordinates, of course, are unique only up to an arbitrary translation, rotation, and reflection, since for any normalized orthogonal matrix Ω , $B = (V\Omega)\Lambda(V\Omega)^T = V\Lambda V^T$.

3. MISSING DATA

The technique of the previous section requires measuring all $n(n - 1)/2$ pair-wise distances. Although this may be feasible for microphone arrays with a small number of microphones (say, $n \leq 10$), it becomes impractical as n increases.

Since without noise B is of rank p , there is much redundancy in B and D . In fact, it can be shown that in a p -dimensional space one need only measure the distances between each point and a set of $p + 1$ basis points, which is $\sum_{k=1}^p k + (p + 1)(n - p - 1)$ measurements. The savings can be significant: 70 measurements instead of 190 for $n = 10$, or 190 measurements instead of 1225 for $n = 50$, with $p = 3$. In this section, we derive a basis for the three-dimensional world (3D) by first looking at cases of 1D and 2D. The derivation is based on the equations found in [7].

3.1. Deriving a 1D basis

Suppose we can measure the distance δ_{AB} between two points A and B . By arbitrarily placing them on the x axis (i.e., $y_A = z_A = y_B = z_B = 0$) and setting the origin as their midpoint we get the following coordinates for the

points:

$$x_A = -x_B = -\frac{1}{2}\delta_{AB},$$

where we have arbitrarily chosen the negative sign.

A and B define a basis in a 1D space. Given the distances δ_{AQ} and δ_{BQ} between A and B and some point Q , the definition of distance gives

$$\begin{aligned} \delta_{AQ}^2 &= (-\frac{1}{2}\delta_{AB} - x_Q)^2 + y_Q^2 + z_Q^2 \\ \delta_{BQ}^2 &= (\frac{1}{2}\delta_{AB} - x_Q)^2 + y_Q^2 + z_Q^2. \end{aligned} \quad (1)$$

Combining these equations and solving for x_Q yields

$$x_Q = \frac{\delta_{AQ}^2 - \delta_{BQ}^2}{2\delta_{AB}}. \quad (2)$$

Thus, Eq. (2) provides the x -coordinate of an arbitrary point Q given the distances between it and the basis points A and B .

3.2. Deriving a 2D basis

If Q does not lie exactly on the line connecting A and B (the x axis), then its residual distance to that line will be nonzero. The square of the residual distance from Q to the x -axis can be computed from Eqs. (1) and (2):

$$\begin{aligned} y_Q^2 + z_Q^2 &= \delta_{AQ}^2 - (-\frac{1}{2}\delta_{AB} - x_Q)^2 \\ &= \frac{1}{2}\delta_{AQ}^2 - \frac{1}{4}\delta_{AB}^2 + \frac{1}{2}\delta_{BQ}^2 - x_Q^2. \end{aligned} \quad (3)$$

Rearranging yields a convenient formula that we will use again:

$$x_Q^2 + y_Q^2 + z_Q^2 = \frac{1}{2}\delta_{AQ}^2 - \frac{1}{4}\delta_{AB}^2 + \frac{1}{2}\delta_{BQ}^2. \quad (4)$$

We may augment the basis with a third point C , where x_C is computed using Eq. (2). By arbitrarily placing C in the xy -plane (i.e., $z_C = 0$), its y -coordinate can be found from Eq. (3):

$$y_C = \sqrt{\frac{1}{2}\delta_{AC}^2 - \frac{1}{4}\delta_{AB}^2 + \frac{1}{2}\delta_{BC}^2 - x_C^2},$$

where we have arbitrarily chosen the positive square root.

A , B , and C now provide a basis for a 2D space. Given the distances δ_{AQ} , δ_{BQ} , and δ_{CQ} for some point Q , the definition of distance

$$\delta_{CQ}^2 = (x_C - x_Q)^2 + (y_C - y_Q)^2 + (z_C - z_Q)^2,$$

by rearranging terms, gives

$$2y_C y_Q = x_C^2 + y_C^2 + x_Q^2 + y_Q^2 + z_Q^2 - \delta_{CQ}^2 - 2x_C x_Q. \quad (5)$$

Applying Eq. (4) to the points C and Q , and substituting back into Eq. (5) yields

$$2y_C y_Q = \frac{1}{2}(\delta_{AC}^2 - \delta_{AB}^2 + \delta_{BC}^2 + \delta_{AQ}^2 + \delta_{BQ}^2) - \delta_{CQ}^2 - 2x_C x_Q.$$

Solving for y_Q ,

$$y_Q = \frac{\delta_{AC}^2 - \delta_{AB}^2 + \delta_{BC}^2 + \delta_{AQ}^2 + \delta_{BQ}^2 - 2\delta_{CQ}^2 - 4x_C x_Q}{4y_C}. \quad (6)$$

Thus, Eqs. (2) and (6) provide the x - and y -coordinates of an arbitrary point Q given the distances between it and the basis points A , B , and C .

3.3. Deriving a 3D basis

Given distances δ_{AD} , δ_{BD} , and δ_{CD} , we may augment the basis with a fourth point D , where x_D and y_D are found using Eqs. (2) and (6), respectively. To find z_D , simply apply Eq. (4) to D , yielding

$$z_D = \sqrt{\frac{1}{2}\delta_{AD}^2 - \frac{1}{4}\delta_{AB}^2 + \frac{1}{2}\delta_{BD}^2 - x_D^2 - y_D^2},$$

where we have again arbitrarily chosen the positive square root.

The points A , B , C , and D now provide a basis for a 3D space. Given the distances δ_{AQ} , δ_{BQ} , δ_{CQ} , and δ_{DQ} for some point Q , the definition of distance for δ_{DQ}^2 gives, after substituting Eq. (4) and solving for z_Q ,

$$\begin{aligned} z_Q = & \frac{1}{4z_D} (\delta_{AD}^2 + \delta_{BD}^2 + \delta_{AQ}^2 + \delta_{BQ}^2 - \delta_{AB}^2 \\ & - 2\delta_{DQ}^2 - 4x_D x_Q - 4y_D y_Q). \end{aligned} \quad (7)$$

Thus, Eqs. (2), (6), and (7) provide the x -, y -, and z -coordinates of an arbitrary point Q given the distances between it and the basis points A , B , C , and D .

4. SIMULATIONS

We tested the approach presented here using simulations in Matlab. We ran three separate experiments, in each case testing the robustness of the algorithm with respect to various types of noise.

The ground truth microphone locations were stored in an $n \times p$ matrix Y , where p is the number of dimensions, while the result of the algorithm was stored in an $n \times p$ matrix X . Since the solution has an arbitrary translation, rotation, and reflection, we measured the root-mean-square (RMS) error in the solution as $\sqrt{R^2}/n$, where

$$R^2 = \sum_{i=1}^n (A^T \mathbf{x}_i - \mathbf{y}_i)^T (A^T \mathbf{x}_i - \mathbf{y}_i),$$

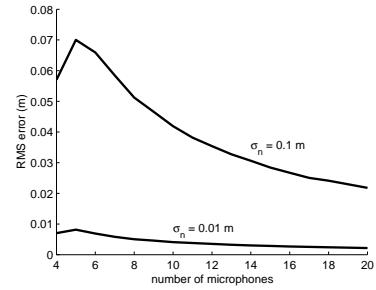


Fig. 2. Classical MDS is accurate and robust with respect to Gaussian measurement noise.

where \mathbf{y}_i is the i th row of Y , \mathbf{x}_i is the i th row of X , and A is the normalized orthogonal matrix that best aligns the points by rotating and reflecting them. (Note that the best translation vector is the zero vector if the centroid of the points is the origin.) The A which minimizes R^2 is given by [2]

$$A = (X^T Y Y^T X)^{\frac{1}{2}} (Y^T X)^{-1}.$$

Note that this approach is simpler than Procrustes' analysis, which also allows for scale changes between the point sets.

4.1. Perturbing distances with Gaussian noise

First we distributed a number of microphones randomly, using a uniform distribution, in a cubic room $5 \text{ m} \times 5 \text{ m} \times 5 \text{ m}$. We then constructed a squared-distance matrix D , perturbed with additive, independent Gaussian noise. The output of the classical MDS algorithm described in Section 2 was evaluated using the error metric described above.

We followed this procedure while varying the number of microphones and the standard deviation of the Gaussian noise. The number of microphones ranged from 4 to 20, and the standard deviation of the noise σ_n was 0.01 m and 0.1 m. The results are displayed in Figure 2, where each data point represents the average over 1000 trials.

For realistic measurement error ($\sigma_n = 0.01 \text{ m}$) possible with even a simple measuring device such as a tape measure, the RMS error is just 7.9 mm with four microphones and less than 4 mm with at least eleven microphones. These errors are significantly smaller than the wavelength of the highest audible frequency, which is approximately 16 mm. When the measurement error is increased by an order of magnitude, the RMS error remains less than 8 cm. Thus, we conclude that the accuracy of classical MDS is well above that required for calibrating microphone arrays.

4.2. Perturbing distances with impulse noise

Unfortunately, the algorithm is not so forgiving with impulse noise, as shown in Figure 3, where each data point

←

→

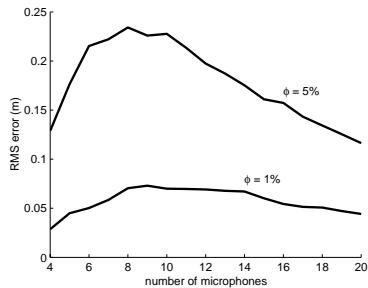


Fig. 3. Classical MDS is sensitive to impulse measurement noise (e.g., missing data).

again represents the average of 1000 trials. This plot shows the RMS error that occurred when a certain percentage ϕ of the distances were set to zero (similar results were achieved with other values) to simulate missing data. With just 1% of the distances missing, the RMS error ranges from 3 cm to 7 cm, and with 5%, it ranges from 12 cm to 24 cm. Clearly, when some data are missing we cannot simply ignore that fact.

4.3. Handling Missing Data

To test our ability to handle the missing data, we ran simulations for $n = 10, 20$, and 50 . For each set of microphones we placed them randomly in space, selected the four best microphones for a basis set (where “best” is defined below), and then filled in missing data using the equations of Section 3. We varied the fraction of missing data from 0 to the maximum allowable for that number of microphones. Plotted in Figure 4 are the results, averaged over 100 trials per data point. The RMS error remains below 5 mm, even when nearly 80% of the measurements are unavailable. These results confirm the statements of [8] and [9] that only about 25-33% of the measurements are needed in practice.

To achieve good results, the basis points should span the space as much as possible, i.e., the volume of the pyramid they define should be as large as possible ([9] gives an alternate measure). The volume of this pyramid can be easily computed as $|v|/6$, where v is the determinant of the following 4×4 matrix:

$$v = \begin{vmatrix} \mathbf{x}_A & \mathbf{x}_B & \mathbf{x}_C & \mathbf{x}_D \\ 1 & 1 & 1 & 1 \end{vmatrix}.$$

5. CONCLUSION

The positions of microphones in an array can be computed using a simple, linear algorithm derived many years ago in the context of multidimensional scaling. The algorithm requires as input the distances between microphones, which

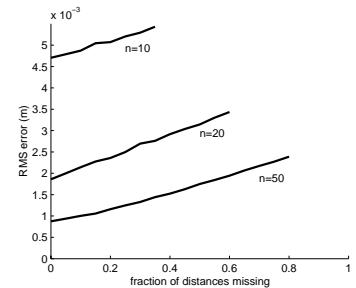


Fig. 4. Classical MDS is robust with respect to missing data when that data is filled in using the technique described in Section 3.

can be measured with a standard tape measure or similar device. We have shown using simulations that the algorithm is robust with respect to Gaussian noise at the level one would expect in practice. We have also presented a technique for handling missing data and have shown that the algorithm is quite robust, even when a large percentage of the measurements are unavailable.

6. REFERENCES

- [1] Jacqueline Meulman, *A Distance Approach to Nonlinear Multivariate Analysis*, Leiden: DSWO Press, 1986.
- [2] Trevor F. Cox and Michael A. A. Cox, *Multidimensional Scaling*, London: Chapman and Hall, 1994.
- [3] Joshua M. Sachar, Harvey F. Silverman, and William R. Patterson III, “Position calibration of large-aperture microphone arrays,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002.
- [4] M. Berger and H. F. Silverman, “Microphone array optimization by stochastic region contraction (SRC),” *IEEE Transactions on Signal Processing*, vol. 39, no. 11, pp. 2377–2386, Nov. 1991.
- [5] G. Young and A. S. Householder, “Discussion of a set of points in terms of their mutual distances,” *Psychometrika*, vol. 3, no. 1, pp. 19–22, Mar. 1938.
- [6] W. S. Torgerson, “Multidimensional scaling: I. theory and method,” *Psychometrika*, vol. 17, no. 4, pp. 401–419, Dec. 1952.
- [7] F. W. Young and N. Cliff, “Interactive scaling with individual subjects,” *Psychometrika*, vol. 37, no. 4, pp. 385–415, Dec. 1972.
- [8] R. A. Girard and N. Cliff, “A Monte Carlo evaluation of interactive multidimensional scaling,” *Psychometrika*, vol. 41, no. 1, pp. 43–64, Mar. 1976.
- [9] R. S. Green and P. M. Bentler, “Improving the efficiency and effectiveness of interactively selected MDS data designs,” *Psychometrika*, vol. 44, no. 1, pp. 115–119, Mar. 1979.