Mapping the Neighborhood of Microtonal Music Scales Using Self-Organizing Maps

Mapping the Neighborhood of Microtonal Music Scales Using Self-Organizing Maps

Kousha Nikkar
kousha.nikkar@studenti.polito.it
Dept. of Electronics and Telecommunications
Politecnico di Torino
Turin, Italy

Matteo Sacchetto
matteo.sacchetto@polito.it
Dept. of Electronics and Telecommunications
Politecnico di Torino
Turin, Italy

Cristina Rottondi
cristina.rottondi@polito.it
Dept. of Electronics and Telecommunications
Politecnico di Torino
Turin, Italy

Abstract

This study introduces a novel method for discovering similarities among music scales by mapping them in a two-dimensional space, leveraging a Self-Organizing Map (SOM). Scales are modelled considering microtonal variations typical of the *Radif* in Iranian music and capturing the role of the *Shahed* note as emotional anchor. Eight distinct scale interval patterns from Western and Persian music theories based on tonic, dominant notes, and microtones are considered in the SOM construction process.

A key application of this mapping is enabling smooth modulation between scales. Using Bresenham's Algorithm, modulation pathways can be computed based on the generated SOM, allowing for gradual transitions that are hardly perceptible by the listener.

A quantitative analysis measured the smoothness of transitions and the coherence of the proposed modulation pathways, considering 40 modulation transitions appearing in classic and contemporary music pieces: a comparison of the modulation pathways, obtained by querying our SOM, to the corresponding scale transitions appearing in the original scores yielded an average hit rate of 0.83. Moreover, a user survey with 79 participants confirmed that the system effectively aligns with traditional modulation practices descending from classical theories of harmony, while introducing innovative transitions.

CCS Concepts

• Computing methodologies → Search methodologies; **Discrete space search**; **• Applied computing → Sound and music computing**; **• Information systems → Proximity search**.

Keywords

Self Organizing Maps, Music Key Modulation, Music Composition

ACM Reference Format:
Kousha Nikkar, Matteo Sacchetto, and Cristina Rottondi. 2025. Mapping the Neighborhood of Microtonal Music Scales Using Self-Organizing Maps. In *20th International Audio Mostly Conference (AM '25), June 30-July 04, 2025, Coimbra, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3771594.3771618

ACM ISBN 979-8-4007-2065-9/25/06
https://doi.org/10.1145/3771594.3771618

1 Introduction

Music is a universal medium for expressing emotions, ideas, and stories, achieved through a combination of melody, harmony, rhythm, and tonal dynamics. Among the many techniques composers use to shape a piece's emotional and structural narrative, modulation (i.e., the transition between musical keys) stands out as an essential tool [18]. This technique allows composers to guide listeners through contrasting emotional landscapes while maintaining harmonic coherence and enhancing the overall complexity of the musical piece.

In any musical work, preserving the tonal integrity of the key is fundamental to creating a harmonious experience [22]. The choice of key defines the tonal center, influencing the emotional character of a piece and the relationships between notes and chords. However, composers often choose to modulate, i.e., to change the key, to deliberately introduce contrast, expand expressive possibilities, and sustain the listeners' interest [21]. Modulation is particularly effective when executed seamlessly, allowing the transition to unfold naturally without disrupting the flow of the composition.

In the context of Persian classical music, modulation is deeply rooted in the tradition of the *Radif*, a structured system of melodic patterns passed down through generations [4]. The *Radif* provides a framework for both composition and improvisation, guiding musicians in their exploration of different tonal spaces. A defining feature of Persian music is its extensive use of Microtones (MTs), which contribute to its rich tonal palette and unique expressive qualities [23]. Modulation often revolves around the *axis note*, known as the *Shahed*, which serves as a melodic anchor and establishes a sense of resolution and emphasis [17]. The *Shahed* is a crucial element in defining the emotional direction of a piece, acting as a reference point for modulation between different scales or modes. Typically, it is the most frequently repeated or longest sustained note in a musical section that does not contain any modulation change, and appears at the end of phrases, acting as a place of melodic resolution.

Traditionally, musicians have relied on intuition and empirical knowledge to navigate Modulation Pathways (MPs, i.e., a sequence of transitions between scales), drawing on their understanding of the relationships between the tonic, dominant, and axis notes of different keys. These relationships are influenced by the intervallic similarities between modes, facilitating smooth and coherent transitions. Despite the significance of modulation in both composition and performance, a systematic analysis of MPs has remained largely unexplored: the absence of a formalized, data-driven approach has left much of the modulations process to the subjective interpretation and experience of musicians.

This research aims at analyzing MPs across Persian and Western classical music scales. The primary objective is to construct a two-dimensional map of musical scales that visually represents their neighborhood relationships and provides a structured framework for modulation. Through Self-Organizing Maps (SOMs), this study clusters scales based on their interval patterns, tonal characteristics, and shared modulation possibilities. This approach enables a systematic exploration of MPs, expanding the expressive possibilities of composers.

A key usage of the SOM is to provide smooth MPs between musical scales, allowing for gradual transitions that are perceived as non-disruptive by the listener. Using Bresenham's Algorithm, MPs can be efficiently generated based on the obtained SOM. To further enhance user understanding, a melody generator was developed using the Python *Mido* library [5], which translates MPs into audible sequences of notes, chosen based on the succession of scales they contain.

After a preliminary analysis to identify the correct dimensioning and parametrization of the SOM, we quantitatively evaluated the smoothness of the MPs generated with our proposed method, by querying the SOM to derive modulation transitions present in already existing musical pieces and comparing the suggested MPs to those appearing in the original scores. Our method achieved an average similarity of 83% between the modulation transitions in the scores and those derived using our approach. Qualitative assessments involved a user survey with 79 participants, who rated 28 generated melodies based on their perceived smoothness of MPs. The results confirmed the system's ability to propose MPs that align with traditional practices while also introducing novel transitions.

In summary, the main contributions of this study are:

- The design of a two-dimensional map of Persian and Western classical music scales using SOMs, capturing their tonal and modulation relationships, to cluster them based on interval patterns, tonal features, and shared modulation potentials,
- The adoption of Bresenham's Algorithm to generate smooth and gradual MPs across the SOM, designed to minimize perceptual disruption.
- The implementation of a Python-based melody generator that converts MPs into audible melodies, facilitating auditory evaluation.

The remainder of the manuscript is organized as follows: Section 2 briefly reviews the related scientific literature, whereas Section 3 summarizes some background notions on the SOMs and the Bresenham's Algorithm. The proposed framework is described in Section 4 and assessed in Section 5. Final remarks are offered in the last Section.

2 Related work

In Western music, modulation techniques such as pivot chords and common-tone transitions have been well studied, but these methods do not easily capture the subtleties inherent to microtonal systems. Persian classical music, with its complex modal structures and the critical role of MTs, presents a distinct approach to modulation. Foundational studies [2, 13, 19] provide detailed analyses of the *Radif* and the modal organization of Persian music, emphasizing how microtonal variations and the central *Shahed* note shape emotional expression and tonal continuity.

However, scientific literature lacks theoretical frameworks specifically devoted to the formalization of the Persian microtonal system: indeed, most of the studies dedicated to Persian music genre classification, mode recognition or composition leverage machine learning models, to bypass the need for analytical representations [8–10, 13]. In this study, we introduce a simple vectorial representation of Persian musical scales, to be used as input to a SOM algorithm.

The seminal work by Kohonen [15] established SOMs as a powerful tool for unsupervised clustering and dimensionality reduction [16, 26]. This methodology has been applied to musical tasks such as content/genre-based clustering and classification. In [11], the authors used a SOM to automatically categorize music files by genre and sonic properties. In [3], SOMs and K-means clustering techniques were used to extract information on the relationship between musical genres. In [1], the authors propose a SOM-based approach for classification of music genres. In [24], the authors take advantage of SOM clustering capabilities to create an assistive music-browsing interface which provides a two-dimensional representation of music collections. In [14], the author introduces a data-driven approach to model the "tonal space" using SOMs. The presented model can take any sound input and project it onto a 2D surface, while following the "tonal trajectories" of chord progressions. In [20], the authors employed SOMs to graphically represent the mood of songs, proposing parameters that describe the emotional content of music.

Recent research has begun to address the specific task of generating MPs, especially in the field of automatic music composition. For instance, González and Santini [12] developed a hybrid method combining numerical analysis and formal grammar to detect modulations, achieving precise identification of modulation points in musical pieces. Differently, our focus is on the generation of MPs, rather than on the identification of modulation points in an existing piece of music.

Brown and George [7] utilized graph theory to explore tonal modulation in popular music, defining modulation graphs based on common scales and analyzing their properties to provide mathematical insights into pivot modulation. By navigating such graphs, MPs can be identified. In our proposed approach, instead, MPs are generated by navigating a SOM.

The above-mentioned studies represent initial steps toward a more systematic and data-driven understanding of MPs in music.

3 Background

In the following we briefly describe the structure of Self Organizing Maps and how Bresenham's algorithm works.

3.1 Self Organizing Maps

SOMs [15] are an unsupervised learning method used for clustering and dimensionality reduction. SOMs map high-dimensional data into a lower-dimensional grid composed by nodes, while preserving topological relationships.

The SOM algorithm consists of the following steps:

(1) *Initialization*: Each node in the grid is associated to a weight vector $\mathbf{w}_i$, which has the same dimension as the first input data vector $\mathbf{x}$. Let the weight vector of node i at time t be denoted as $\mathbf{w}_i(t)$. Its values are initially chosen randomly from a predefined range.
(2) *Competition Phase*: The next step is to determine the Best Matching Unit (BMU), i.e., the node whose weight vector $\mathbf{w}_i$ is closest to the input vector $\mathbf{x}$. The distance between the input vector $\mathbf{x}$ and each node's weight vector $\mathbf{w}_i$ is computed, typically using the Euclidean distance[1]:

$$d(\mathbf{x}, \mathbf{w}_i) = \|\mathbf{x} - \mathbf{w}_i\|_2$$

The BMU i_{bm} is the node that minimizes this distance:

$$i_{\text{bm}} = \arg\min_i d(\mathbf{x}, \mathbf{w}_i)$$

(3) *Adaptation Phase*: After identifying the BMU i_{bm}, the next step is to update the weight vectors of the BMU and of its neighboring nodes through the following formula:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \eta(t) h_{i,i_{\text{bm}}}(t)(\mathbf{x} - \mathbf{w}_i(t))$$

where $h_{i,i_{\text{bm}}}(t)$ is the neighborhood function that determines the influence of the BMU on neighboring nodes and $\eta(t)$ is the learning rate at time t.
The neighborhood function $h_{i,i_{\text{bm}}}(t)$ typically decreases over time, meaning that the influence of the BMU on neighboring nodes becomes smaller as the algorithm progresses. A common form of $h_{i,i_{\text{bm}}}(t)$ is:

$$h_{i,i_{\text{bm}}}(t) = \exp\left(-\frac{d(i, i_{\text{bm}})^2}{2\sigma(t)^2}\right)$$

where $\sigma(t)$ is the neighborhood radius and $d(i, i_{\text{bm}})$ is the Euclidean distance between the node i and the BMU i_{bm} in the grid structure.
(4) *Iterative Process*: The SOM algorithm proceeds iteratively. In each iteration, a new input vector $\mathbf{x}$ is presented to the network, and the BMU and weight updates are computed. Over time, $\eta(t)$ and $\sigma(t)$ decrease, and the map becomes more refined. $\eta(t)$ and $\sigma(t)$ are typically updated as follows:

$$\eta(t) = \eta(0) \exp\left(-\frac{t}{\tau_\eta}\right)$$
$$\sigma(t) = \sigma(0) \exp\left(-\frac{t}{\tau_\sigma}\right)$$

where $\eta(0)$ is the initial learning rate, $\sigma(0)$ is the initial neighborhood radius, t is the current time step, τ_η is a constant that controls the rate of decay of $\eta(t)$ and τ_σ is a constant that controls the rate of decay of $\sigma(t)$. However, in the implementation used in this paper, both $\eta(t)$ and $\sigma(t)$ are inverse time decay functions that decay based on the number of iterations of the algorithm.
(5) *Termination*: Typically, the SOM algorithm continues until the weight vectors converge, meaning that the positions of the nodes no longer change significantly with additional iterations. The convergence criterion can be expressed as:

$$\sum_{i=1}^{N} \|\Delta\mathbf{w}_i(t)\|_2 < \varepsilon$$

where $\Delta\mathbf{w}_i(t) = \mathbf{w}_i(t+1) - \mathbf{w}_i(t)$ is the weight change of the i-th node at time t, and ε is a small, positive value that acts as a threshold. The choice of ε affects the stopping point of the algorithm, as it determines the total weight change deemed sufficiently small to guarantee convergence. A smaller ε enforces a stricter convergence, potentially requiring longer training times.
However, selecting an appropriate and optimal value for ε is often challenging. Therefore, in practice, an alternative, simpler approach is frequently adopted for termination of the SOM training, i.e., stopping after a predefined, fixed number of iterations (NI). This approach, while not necessarily guaranteeing convergence, provides a predictable training duration and can be adequate for many applications, especially when computational resources are limited or a map with a coarser level of detail is acceptable. For the specific SOM implementation detailed in this paper, this iteration-based termination method was chosen for its simplicity and practicality.

Once the algorithm terminates, the SOM map can be visualized as a two-dimensional grid, where each node represents a cluster of input vectors. The organization of the grid reflects the relationships between data points, where similar input vectors are mapped onto adjacent nodes.

3.2 Bresenham's Algorithm

Bresenham's Algorithm [6] is a highly efficient algorithm used to draw straight lines on a raster grid. It is an integer-based algorithm that minimizes the need for floating-point operations, making it particularly useful in computer graphics, especially on devices with limited processing power.

The algorithm comprises the following steps:

- *Initialization*: Given two points (x_0, y_0) and (x_n, y_n) that define the line vertexes on the raster grid, determine the absolute differences in the x and y coordinates:

$$\Delta_x = |x_n - x_0|$$
$$\Delta_y = |y_n - y_0|$$

Compute the step directions:

$$x_{\text{step}} = \text{sign}(x_n - x_0)$$
$$y_{\text{step}} = \text{sign}(y_n - y_0)$$

Initialize the decision parameter:

$$p_0 = \begin{cases} 2\Delta_y - \Delta_x & \text{if } \Delta_x \geq \Delta_y \\ 2\Delta_x - \Delta_y & \text{if } \Delta_x < \Delta_y \end{cases}$$

- *Iterative Process*: Starting from (x_0, y_0), for each step $k = 0, 1, 2, \ldots, n-1$, compute the next point (x_{k+1}, y_{k+1}) based on the current value of the decision parameter p_k. Then, compute the next value of the decision parameter p_{k+1}. Repeat the process until (x_n, y_n) is reached.

[1] In this work, Euclidean distance is used as it provides a direct geometric interpretation. Future work will be devoted to a comparative assessment of alternative distance measures, e.g. circular-aware metrics, which could capture more effectively the cyclic and perceptually non-linear properties of pitch-class vectors.

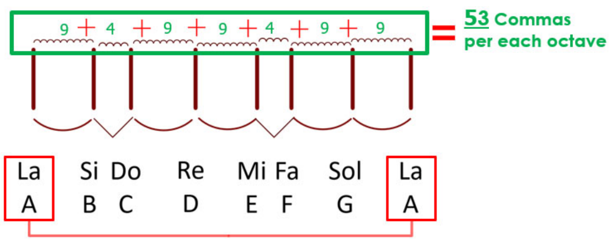

Figure 1: Representation of an octave and of its constituting microtonal intervals.

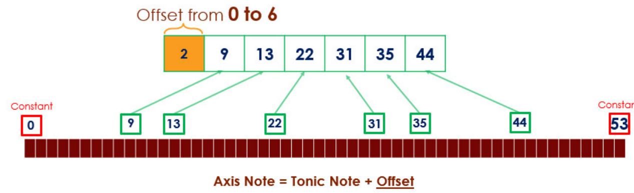

Figure 2: Representation of the encoding of the A minor musical scale, using C an axis note, i.e., with an offset of two with respect to the tonic note.

Depending on the primary direction (horizontal or vertical), the iterative process follows one of two distinct implementations.

Case 1: Primary direction is horizontal ($\Delta_x \geq \Delta_y$)

$$x_{k+1} = x_k + x_{\text{step}}$$

$$y_{k+1} = \begin{cases} y_k & \text{if } p_k < 0 \\ y_k + y_{\text{step}} & \text{if } p_k \geq 0 \end{cases}$$

$$p_{k+1} = \begin{cases} p_k + 2\Delta_y & \text{if } p_k < 0 \\ p_k + 2(\Delta_y - \Delta_x) & \text{if } p_k \geq 0 \end{cases}$$

If $p_k < 0$, the next point is selected horizontally, otherwise diagonally.

Case 2: Primary direction is vertical ($\Delta_x < \Delta_y$)

$$y_{k+1} = y_k + y_{\text{step}}$$

$$x_{k+1} = \begin{cases} x_k & \text{if } p_k < 0 \\ x_k + x_{\text{step}} & \text{if } p_k \geq 0 \end{cases}$$

$$p_{k+1} = \begin{cases} p_k + 2\Delta_x & \text{if } p_k < 0 \\ p_k + 2(\Delta_x - \Delta_y) & \text{if } p_k \geq 0 \end{cases}$$

If $p_k < 0$, the next point is selected vertically, otherwise diagonally.

- *Termination*: The algorithm continues until the current point (x_k, y_k) becomes equal to the final point (x_n, y_n), after approximately $\max(\Delta_x, \Delta_y)$ steps. The set of points (x_0, y_0), $(x_1, y_1), \ldots, (x_n, y_n)$ approximate the straight line from (x_0, y_0) to (x_n, y_n) on the raster grid.

4 The Proposed Methodology

This Section presents our proposed data representation structure and the method adopted to generate MPs leveraging SOMs.

4.1 Data Representation

As depicted in Fig. 1, we assume a 53 Tone Equal Temperament (53-TET) system, i.e., an octave is subdivided in 53 equal MT intervals, also called *commas*. Each MT is approximately 22.64 cents wide, compared to 100 cents for a Western semitone. It is worth mentioning that, in Persian music, a note can exist in five different alterations: *sharp, flat, natural, koron, and sori*, depending on their position within the 53-TET range.

Based on such representation, we can encode a musical scale using a vector of 7 elements. Any musical scale is assumed to assign to the tonic note a fixed value of 0. Then, the 6 rightmost vector elements correspond to the offsets (expressed in number of MTs) from the tonic note to the other notes (also called *degrees*) of the scale, as represented in Fig. 2. It follows that the numerical values of the vector elements correspond to the pitch relationships between notes. For example, if the second degree (i.e., the supertonic) has an index of 9 and the third degree (i.e., the mediant) has an index of 13, there is a distance of 4 MTs (equivalent to a semitone) between them. Finally, the first element of the vector represents the offset of the axis note with respect to the tonic note, expressed in number of degrees (i.e., in a range from 0 to 6). For example, an offset of 0 indicates that the axis note corresponds to the tonic note of the scale, whereas an offset of 2 denotes as axis note the mediant of the scale.

4.2 Data Collection

The dataset used for creating the SOM includes various scales, each with its specific interval pattern. These scales were chosen based on various musical theory sources, including both Western and Persian classical music literature, as well as contemporary analyses of microtonal music systems. The scales selected for the study include:

- *Major Scale (Mahur)*: This scale follows the common Western pattern of major scales, which serves as the cornerstone of Western tonal music theory.
- *Melodic Minor Scale*: This scale has distinct ascending and descending forms and is used in both Western and Iranian music, highlighting its flexibility and adaptability across cultures.
- *Harmonic Minor Scale*: Known for its peculiar sound, especially in Persian music, this scale has an amplified seventh interval that gives it a dramatic, otherworldly quality.
- *Natural Minor Scale*: This scale follows the diatonic pattern common in Western music.
- *Homayoun Mode*: One of the most important modes in Iranian classical music, known for its distinct emotional quality, often evoking feelings of peace and nostalgia.
- *Shur Mode*: This Persian mode is characterized by a mixture of minor and major intervals and is often used to express melancholy, creating a complex emotional tone.
- *Chahargah Mode*: This mode from Persian classical music evokes a heroic and dramatic tone, frequently used to convey strength, courage, and passion.

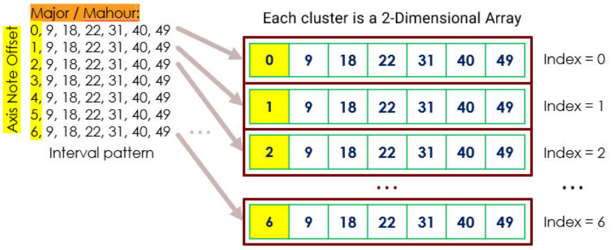

Figure 3: A cluster containing seven Major (Mahour) scales with different axis notes.

- *Segah Mode*: A mode used in Persian classical music that creates a mystical and solemn atmosphere, often associated with spiritual or introspective themes.

Based on the scale representation discussed in the previous subsection, we can create clusters of scales using the same mode, but different axis notes. As an example, Fig. 3 reports a cluster of seven major scales with 7 distinct axis notes.

4.3 SOM Design

4.3.1 Map Initialization. We assume that each node in the grid represents a scale, and its position reflects the proximity relationship to other scales, based on their interval patterns. When initially populating the SOM, we deliberately placed as neighbors those nodes that share the same scale mode but have different axis notes, with the aim of facilitating the learning process. After populating the SOM, all numerical values were normalized within the $[0, 1]$ range.

4.3.2 Weights Initialization. The initial weights of each node in the SOM were set randomly, but within a carefully defined range that reflected the potential values of musical intervals. These intervals were chosen to encompass the full range of possible pitch relationships found in both Western and Persian musical systems. Western music typically operates within a twelve-tone temperament framework, where basic intervals, consisting of whole tones and semitones, constitute the basis of most musical structures. In contrast, Persian music, with its deep history of microtonality, utilizes intervals that fall between Western semitones, resulting in a unique system of MT steps. The inclusion of these MT intervals in the initial range of weights was essential to ensure that the SOM could accommodate the complexities of Persian music.

Table 1: Admissible values for the initialization of the elements of a weight vector $w_i(0)$.

$w_{i,1}$	[0,6]
$w_{i,2}$	$\{4, 6, 9, 12, 14\}$
$w_{i,3}$	$\{13, 15, 18, 21, 23\}$
$w_{i,4}$	$\{17, 19, 22, 25, 27\}$
$w_{i,5}$	$\{26, 28, 31, 34, 36\}$
$w_{i,6}$	$\{35, 37, 40, 43, 45\}$
$w_{i,7}$	$\{44, 46, 49, 52, 54\}$

Therefore, the weight of each node in the SOM corresponds to the potential pitch distances present in the analyzed scale, ensuring that nodes could represent different relationships between internal substructures of both Western and Persian scales. The possible choices for the value of each element of the weight vectors are reported in Table 1.

4.3.3 SOM Implementation. In this paper, we used the *MiniSom* Python library [27] to implement the SOM algorithm. For reference, we report here the default decay functions for $\eta(t)$ and $\sigma(t)$ implemented in that library:

$$\eta(t) = \frac{\eta(0)}{1 + \frac{100t}{\mathrm{NI}}}$$

$$\sigma(t) = \frac{\sigma(0)}{1 + \frac{t(\sigma(0)-1)}{\mathrm{NI}}}$$

4.4 Modulation Pathways Generation

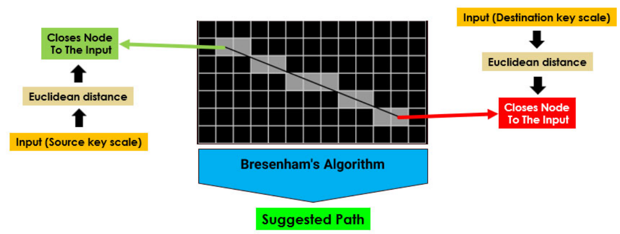

Figure 4: Finding the shortest path between two nodes on the Self-Organizing Map using Bresenham's Algorithm.

Once the SOM is trained, the next step is to generate an MP connecting a starting scale to an ending scale by interconnecting their corresponding nodes with a straight line in the SOM. The MP generation process involves the following steps:

(1) *Find the closest node to the starting scale*: Similar to the process of finding the BMU, we identify the node in the map that is closest to the starting scale and store its index.
(2) *Find the closest node to the ending scale*: We repeat the same process for the ending scale, identifying its closest node in the map and storing its index.
(3) *Apply the Bresenham algorithm*: Using the Bresenham algorithm, we determine the nodes that lie along the straight line path between the source node and the destination node.
(4) *Generate the MP*: The scales represented by the nodes along the line path between the source and destination nodes are suggested as MP, in the order of traversal from starting to ending scale. To ease interpretation, each of the vectors associated to the nodes belonging to the MP is denormalized, and each vector element is converted into a musical symbol (i.e., a note and an associated alteration).

The above-described pipeline is summarized in Fig. 4. It is worth noting that, when displaying the MP scales, many of them may be repeated consecutively in the suggested list. We use the frequency of these repetitions in relation to the total path length to compute and suggest an appropriate duration for each scale in the modulation

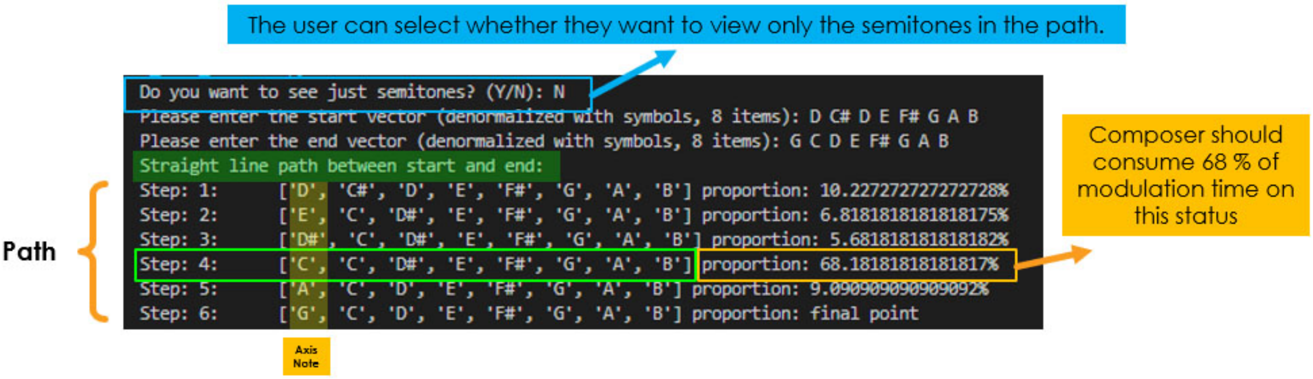

Figure 5: Snapshot of the developed interactive text-based interface.

process, relative to the total desired duration of the transition from starting to ending scale. The duration assigned to each scale in the modulation process is determined by the ratio of the number of repetitions of its corresponding node in the MP to the total number of nodes in the MP.

To facilitate user interaction with the SOM, a simple interactive text-based interface has been developed. As depicted in Fig. 5, the text-based interface requests the user to prompt the starting and ending scale, with notes ordered from C to B, along with the axis note (first element of the list), and displays the intermediate scales and axis notes included in the calculated MP, along with the suggested relative durations.

Moreover, a melody generator was implemented to automatically create a melodic line based on a given MP. This implementation leverages the Python *Mido* library [5] to generate files in Musical Instrument Digital Interface (MIDI) format. The notes that form the melody are chosen randomly within the set of nodes constituting the scales in the MP, with each note's duration set to one quarter note. The transition timings from one scale to the next one are computed based on the suggested durations. The only limitation of this library is its inability to easily generate *koron* and *sori* alterations, so it was necessary to restrict the melody generation process to scales including only semitone intervals (i.e., according to the Western tradition)[2].

5 Numerical Assessment

In this Section, we present a quantitative and qualitative evaluation of our SOM-based MPs generation method.

5.1 Evaluation Criteria

For our quantitative performance assessment, we adopted the following evaluation criteria:

- **Quantization Error (QE)**: it quantifies the average distance between the input vectors and their corresponding BMU on the map. The QE provides insight into how accurately the SOM maps high-dimensional data into a lower-dimensional grid. A lower QE suggests that the map's nodes are closely matching the input vectors, while a higher error indicates that the map is poorly representing the input space and might require improvements. By minimizing the QE, the SOM is able to better represent the tonal relationships between musical scales, which is crucial for generating smooth modulation paths. The QE is defined as follows:

$$\text{QE} = \frac{1}{N}\sum_{i=1}^{N} d(\mathbf{x}_i, \mathbf{w}_{i_{\text{bm}}}) = \frac{1}{N}\sum_{i=1}^{N} \left\|\mathbf{x}_i - \mathbf{w}_{i_{\text{bm}}}\right\|_2$$

where N is the total number of input vectors, $\mathbf{x}_i$ is the i-th input vector and $\mathbf{w}_{i_{\text{bm}}}$ is the weight vector of the BMU of $\mathbf{x}_i$.

- **Topographic Error (TE)**: this metric evaluates how well the SOM preserves the topological relationships between input vectors during training. In the context of musical scale analysis, this is crucial: for modulation paths between musical scales to be smooth and musically meaningful, the map should place similar scales near each other on the grid. Formally, for each input vector $\mathbf{x}_i$, we identify its BMU and its *second-best matching unit (second BMU)*, i.e., the two SOM nodes whose weight vectors are closest to $\mathbf{x}_i$ in terms of Euclidean distance, in the input space. The topographic error for $\mathbf{x}_i$, denoted $\Gamma(\mathbf{x}_i)$, is defined as:

$$\Gamma(\mathbf{x}_i) = \begin{cases} 0 & \text{if the BMU and second BMU are adjacent} \\ 1 & \text{otherwise} \end{cases}$$

Note that two nodes are considered adjacent if they are immediate neighbors in the SOM's grid topology (e.g., horizontally or vertically adjacent in a 2D rectangular grid). The overall TE is then computed as the average of $\Gamma(\mathbf{x}_i)$ across all input vectors:

$$\text{TE} = \frac{1}{N}\sum_{i=1}^{N} \Gamma(\mathbf{x}_i)$$

- **Reconstruction Percentage (RP)**: it evaluates how accurately the map represents the positions of the input vectors in the output space, after the training process has been completed. In the context of musical scales, this is particularly

[2]Future work will be devoted to the integration of microtonal variations via pitch bending, similarly to what already done in [25].

important, as the SOM must retain the key structural features of the input vectors (such as interval patterns and tonal relationships) to ensure that the MPs generated are musically meaningful and coherent. The higher the RP, the more faithfully the map has preserved the input data. The RP is defined as follows:

$$\text{RP} = 100 \times \frac{N_{\text{acc}}}{N}$$

where N_{acc} represents the number of accurately reconstructed input vectors, and N represents the total number of input vectors in the dataset. The accurately reconstructed input vectors are those for which the output map provides a match with the original vector in terms of position or structure. To compute N_{acc}, we employ an exploration algorithm after SOM training. This algorithm iterates through each unique musical scale from the input dataset. For each input vector, we then examine all nodes in the trained SOM map to determine if that specific scale is present in the map's nodes. If at lease one match is found for a given scale, it signifies that the map has successfully reconstructed that scale, and N_{acc} is incremented by one.

5.2 SOM Parameters Tuning

As preliminary analysis, we systematically varied the parameters of the SOM model, including map dimensions, $\sigma(0)$, $\eta(0)$, and NI to define suitable ranges for a fine-grained grid search, with the aim of identifying the optimal configuration that maximizes performance across the three evaluation metrics: QE, TE, and RP.

In the first set of experiments, we trained the SOM with $N = 56$ input scales (i.e., major, harmonic minor, melodic minor, and natural minor scales from Western music, as well as Shur, Homayoun, Segah, and Chahargah, each one replicated using 7 different axis notes), and we focused on varying $\sigma(0)$ and $\eta(0)$, while keeping the map dimensions constant at 60×60 and setting NI $= 10,000$. A higher $\sigma(0)$ results in a wider neighborhood and more significant weight adjustments, which may lead to faster convergence. In contrast, a lower $\sigma(0)$ limits the neighborhood's influence and allows for more fine-grained adjustments to individual neurons, potentially leading to a more precise map representation.

Results obtained by setting $\eta(0) = 0.9$ and varying $\sigma(0)$ are reported in Tab. 2 and show that a significant improvement in QE and RP is achieved when reducing $\sigma(0)$ from 20 to 5, at the price of a slight deterioration of TE. Therefore, we identified the range [6, 10] for a more fine-grained exploration.

Table 2: Impact of the value of $\sigma(0)$ on QE, TP, TR

$\sigma(0)$	**50**	**20**	**5**
QE	0.321	0.226	0.099
TE	0.061	0.071	0.111
RP	2%	6%	45%

Similarly, by setting $\sigma(0) = 10$ we varied the value of $\eta(0)$ to assess how quickly the SOM converged and how well it captured the underlying structure of the input data. Results reported in Tab. 3 shows that reducing $\eta(0)$ from 0.9 to 0.1 leads to minor improvements in TE while slightly degrading QE and RP. Thus, we decided to set $\eta(0) = 0.9$ in all the subsequent experiments.

Table 3: Impact of the value of $\eta(0)$ on QE, TP, TR

$\eta(0)$	**0.9**	**0.5**	**0.1**
QE	0.165	0.163	0.171
TE	0.101	0.086	0.056
RP	17%	16%	14%

The second set of experiments focused on varying the size of the SOM. Note that the theoretical space of possible scales to be used as input for the SOM construction is $7 \times 5^6 = 109,375$, where 7 is the number of possible axis note offsets and 5 is the number of alterations for each of the subsequent 6 notes. Since $\sqrt{109,375} \approx 330$, we empirically set an upper bound to the search range to 400 nodes. During this phase, the parameters $\sigma(0)$, $\eta(0)$, and NI are kept constant ($\sigma(0) = 10$, $\eta(0)$=0.9, and NI=10,000).

To assess the impact of the SOM dimensions on the three evaluation criteria, we first normalized their values in the [0, 1] range (indicated in the following as NTE, NQE, and NRP). To maintain consistency across all evaluation criteria, where lower values indicate better performance, we normalized RP using its 1-complement. This is because, unlike TE and QE, lower RP values denote poorer performance.

Intuitively, larger maps can capture finer details and provide a more precise representation, particularly beneficial for complex musical structures like microtonal scales. However, this comes at the cost of longer training times, increased computational cost and more complex parameter tuning. In contrast, smaller maps train faster, require fewer computational resources, and a simpler parameter adjustments. However, they lack the granularity needed to preserve subtle relationships between scales.

Fig. 6 plots the trend of the normalized TE, QE, and RP as a function of the SOM size. Given the abrupt increase in RP and QE occurring around size 240×240 and the steep decrease of QE around size 250×250, we opted to further refine the choice of the map size by shrinking the search range to [236, 245].

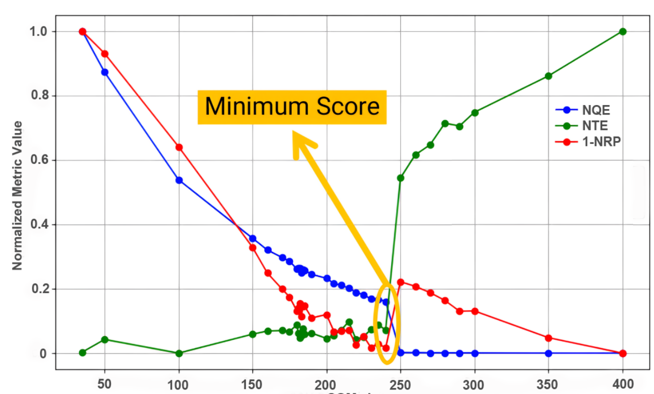

Figure 6: Impact of map dimensions on criteria: TE, QE, and RP.

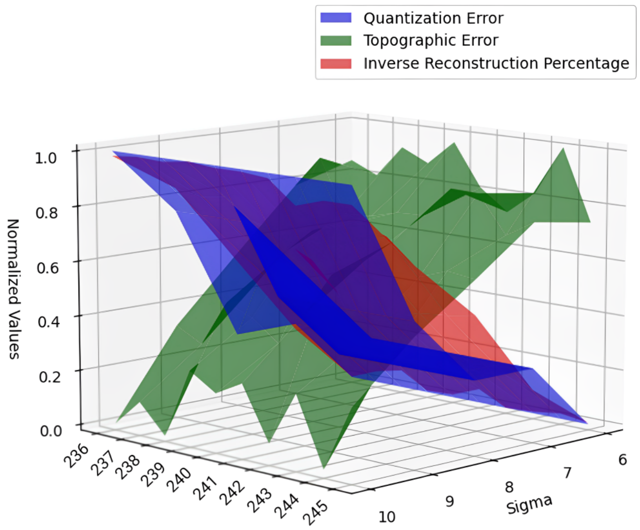

Figure 7: 3D visualization of the grid search results for SOM size and $\sigma(0)$, visualizing the trade-off among all three evaluation criteria found by minimizing their summed normalized values.

Thus, we further refined the tuning of the map size and $\sigma(0)$ by conducting a systematic grid search, while setting $\eta(0) = 0.9$ and NI $= 10,000$. To assess the trade-off between the three evaluation criteria, we computed the sum of the normalized values, giving equal weighting to each criterion. This sum represented the cumulative performance of the SOM across all criteria.

$$S = \alpha \cdot \text{NQE} + \beta \cdot \text{NTE} + (1 - \alpha - \beta) \cdot (1 - \text{NRP})$$

where NQE, NTE and NRP are the normalized versions of QE, TE and RP for a given configuration. The goal was to identify a configuration that minimized the sum of the normalized values (S), as this would indicate the best balance across the three criteria. Based on the results reported in Fig. 7, we concluded that S was minimized by setting $\sigma(0) = 6$ and the SOM size to 242×242.

Finally, we investigated the impact of NI on QE, TE and RP, using the above-identified parameters configuration. As reported in Fig. 8, $200,000$ iterations are sufficient to achieve extremely low QE and almost perfect RP, while obtaining a marginal improvement of TE with respect to stopping the execution of the SOM algorithm after $150,000$ iterations. Therefore, we set NI $= 200,000$.

It follows that the final SOM configuration consisted of a map dimension of 242×242, a $\sigma(0) = 6$, an $\eta(0) = 0.9$, and NI$=200,000$.

5.3 Evaluation of Modulation Pathways

The quality of the generated MPs was assessed through both objective and subjective evaluation methods.

5.3.1 Quantitative Assessment. We queried the trained SOM with 50 starting/ending scale pairs, randomly chosen among conventional scales widely adopted in classical and contemporary music pieces. For each of the obtained MPs, we considered the sequence of scales included in it and calculated the differences between each pair of consecutive scales to quantify the smoothness of the transition. Specifically, we calculated the sum of the absolute difference (in number of MTs) between each pair of corresponding notes in the pair of consecutive scales. Fig. 9 plots the distribution of such differences. In general, we have an average difference of 5 MTs, which suggests that on average between each pair, 3 notes remain identical and 5 change by a MT. Since two scales may differ at least by one MT, this results suggests that the generated MPs are relatively smooth.

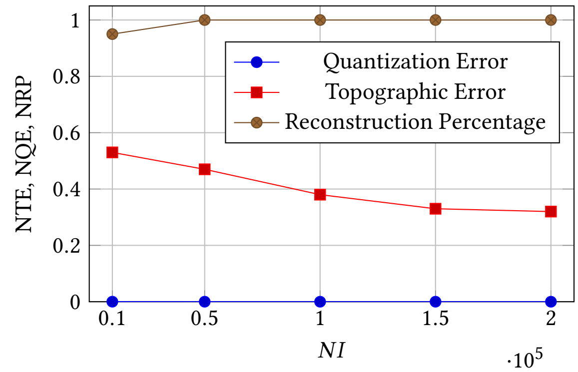

Figure 8: Trend of normalized QE, TE and RP as a function of the number of iterations, NI.

Furthermore, we considered 40 musical pieces and identified modulation transitions in their scores. For each transition, we recorded starting and ending scales in a list, together with their axis notes, identified as the most frequent notes in their respective musical section (assuming that a section includes a set of consecutive bars where no modulations occur). We then queried the SOM with the 40 obtained scale pairs, obtaining 40 MPs. Finally, we compared the MPs generated by SOM to the succession of scales appearing in the music scores and, for each scale within the score, we recorded whether the same scale appeared in the MP. We calculated the Hit Rate (HR) by computing the ratio of the number of scales included in the MP that also appeared in the music score to the total number of scales in the MP.

$$\text{HR} = \frac{\text{Nr. of scales in the MP appearing in the original score}}{\text{Nr. of scales in the MP}}$$

The average HR was 0.83. It is worth noting that the SOM was specifically designed to focus on gradual changes, whereas composers often deliberately included abrupt modulations in their pieces. Therefore, HR values close to 1 are hardly expectable.

5.3.2 Qualitative Assessment. To further validate the effectiveness of our proposed method, a web-based survey was conducted where participants were asked to listen to 28 melodies generated by our automatic melody generator, based on 28 MPs produced by querying the SOM, and to rate them based on the perceived smoothness of the modulation. Each melody contained between 5 and 20 scale changes, with 30 quarter-note tones in between each transition. Thus, the duration of the excerpts varied from 25 to 180 seconds.

The objective was to evaluate how smoothly the key or mode changes occurred within a piece, and whether these transitions were noticeable to the listener. After listening to each of the provided

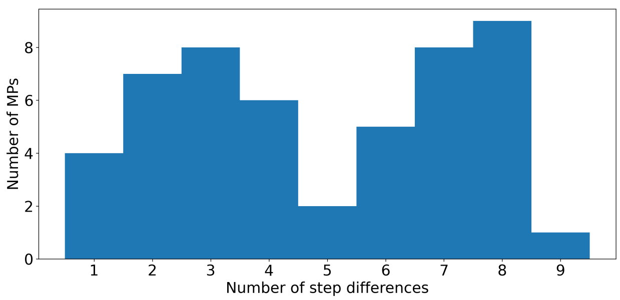

Figure 9: Distribution of MPs by number of step differences.

files, the participants were asked to rate their perception of the modulation shifts on a Likert scale from 1 to 5, where 1 meant that the participants clearly noticed the scale changes and 5 meant that the changes were deemed not recognizable. If unsure, the participants also had the possibility to leave it blank. In total, 79 subjects participated in this survey, and each melody received a number of ratings between 62 and 72. The age of the respondents was in the 22-55 years range. Except for a few professionals in the music field, the majority of the participants were either amateur musicians or had no musical education.

Fig. 10 shows the average rating obtained by each of the 28 melodies. Overall, the algorithm proves to be capable of generating transitions perceived as at least moderately smooth for a significant portion of MPs, as indicated by the average Likert scale ratings, generally exceeding 3. However, substantial variability in these ratings across different MPs suggests that smoothness is not uniformly perceived, and that certain transitions are perceived as more abrupt than others. This could be due to multiple factors.

Firstly, it may be related to the intrinsic characteristics of the selected scale transitions. For instance, transitions achieving the highest average smoothness scores (above 4.7) include DM-GM, DM-F#m, CM-FM, Am-Dm, and CM-Am. DM-GM and CM-FM involve transitions between major keys a perfect fourth or fifth apart, while Am-Dm represents a similar relationship in minor keys. CM-Am is a relative major to minor transition, also scoring highly. Conversely, transitions achieving lower smoothness ratings (below 3.1) include AM-C#m, Cm-BbM, BbM-G#m, Em-DM, and G#m-F#M. Interestingly, a pattern emerges in this group, with several transitions involving a shift towards a major key from a minor or altered key (Cm-BbM, G#m-F#M, Em-DM). The transition AM-C#m and BbM-G#m involve mode shift as well or move to altered minor key respectively.

Secondly, the ratings variability may be related to amount of key changes in a melody. As previously mentioned, each melody may contain between 5 and 20 scale changes. Thus, MPs having a high number of transitions may be intrinsically biased towards a lower rating, since the chance that at least one of such transitions is perceived as noticeable increases with their number.

6 Conclusion

This study introduced a novel approach to generating modulation pathways between musical scales using Self-Organizing Maps (SOMs). By clustering scales based on their interval patterns, including microtonal variations, the method effectively captured structural relationships in both Western and Persian music traditions.

The generated modulation pathways were evaluated using both objective metrics and subjective feedback from human listeners: the system was tested against 40 real musical pieces, achieving an 83% hit rate in predicting scale transitions. Additionally, a user survey involving 79 participants, who rated 28 generated melodies based on modulation pathways produced by the system, confirmed that the modulation transitions were perceived as smooth and aligned with conventional traditional practices.

These results highlight the potential of computational models to facilitate seamless modulations, offering new tools for composers and music theorists to explore novel pathways between tonal spaces, while seamlessly blending elements from different musical traditions.

Acknowledgments

We acknowledge the support of the "Musical Metaverse: an inclusive Extended Reality platform for networked musical interactions" project (grant n. 2022CZWWKP) – funded by European Union – Next Generation EU within the PRIN 2022 program (D.D. 104 - 02/02/2022 Ministero dell'Università e della Ricerca). This manuscript reflects only the authors' views and opinions and the Ministry or the European Union cannot be considered responsible for them.

References

[1] Abdul Nafey Ahmad, Chandra Sekhar, and Abhinav Yashkar. 2014. Music Genre Classification Using Music Information Retrieval and Self Organizing Maps. In *Proceedings of the Third International Conference on Soft Computing for Problem Solving*, Millie Pant, Kusum Deep, Atulya Nagar, and Jagdish Chand Bansal (Eds.). Springer India, New Delhi, 625–634.

[2] Siavash Akhlaghi. 2013. *Modulating the Logics of Iranian Classical Music*. Master's thesis. Institute of Sonology, Royal Conservatoire The Hague. https://sonology.org/wp-content/uploads/2019/10/Thesis-SiavashAkhlaghi-Sonology-2013.pdf

[3] A. Azcarraga and F. K. Flores. 2017. A Study on Self-Organizing Maps and K-Means Clustering on a Music Genre Dataset. In *Theory and Practice of Computation*. World Scientific, Singapore, 219–234. doi:10.1142/9789813234079_0017

[4] Carol M Babiracki and Bruno Nettl. 1987. Internal Interrelationships in Persian Classical Music: The Dastgah of Shur in Eighteen Radifs. *Asian music* 19, 1 (1987), 46–98.

[5] Ole Martin Bjørndalen. 2025. Mido: MIDI Objects for Python. https://mido.readthedocs.io/ Version 1.3.2.

[6] J. E. Bresenham. 1998. *Algorithm for computer control of a digital plotter*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/280811.280913

[7] Jason I. Brown and Ian C. George. 2024. Modulation Graphs in Popular Music. *The Mathematical Intelligencer* 46, 4 (Dec. 2024), 332–339. doi:10.1007/s00283-023-10325-y

[8] Mansoure Ebrahimi, Babak Majidi, and Mohmmad Eshghi. 2019. Procedural Composition of Traditional Persian Music Using Deep Neural Networks. In *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*. IEEE, New York, NY, USA, 521–525. doi:10.1109/KBEI.2019.8734959

[9] Danial Ebrat, Farzad Didehvar, and Milad Dadgar. 2022. Iranian Modal Music (Dastgah) detection using deep neural networks. arXiv:2203.15335 [cs.SD] https://arxiv.org/abs/2203.15335

[10] Nacer Farajzadeh, Nima Sadeghzadeh, and Mahdi Hashemzadeh. 2023. PMG-Net: Persian music genre classification using deep neural networks. *Entertainment Computing* 44 (2023), 100518. doi:10.1016/j.entcom.2022.100518

[11] Markus Frühwirth and Andreas Rauber. 2002. Self-Organizing Maps for Content-Based Music Clustering. In *Neural Nets WIRN Vietri-01*, Roberto Tagliaferri and Maria Marinaro (Eds.). Springer London, London, 228–233.

[12] Maria Rojo González and Simone Santini. 2019. Structural characterization of musical harmonies. arXiv:1912.12362 [cs.MM] https://arxiv.org/abs/1912.12362

[13] Peyman Heydarian. 2016. *Automatic recognition of Persian musical modes in audio musical signals*. Ph. D. Dissertation. London Metropolitan University. https://repository.londonmet.ac.uk/1190/1/HeydarianPeyman%20-%20%

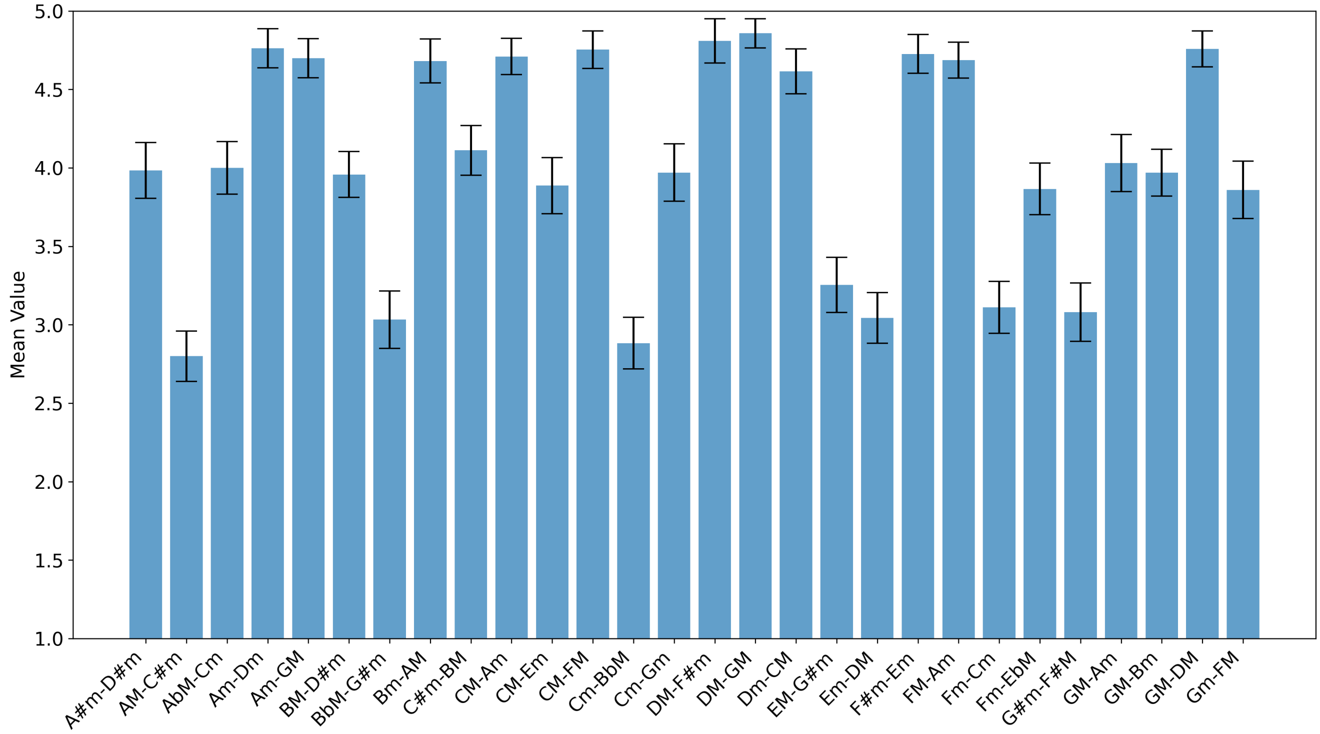

Figure 10: Average perceived smoothness of MPs, with 95% confidence intervals.

20PhD%20Full%20Thesis.pdf

[14] Petr Janata. 2007. Navigating Tonal Space. *Computing in Musicology* 15 (2007), 39–50.

[15] T. Kohonen. 1990. The self-organizing map. *Proc. IEEE* 78, 9 (1990), 1464–1480. doi:10.1109/5.58325

[16] Teuvo Kohonen. 2013. Essentials of the self-organizing map. *Neural Networks* 37 (2013), 52–65. doi:10.1016/j.neunet.2012.09.018 Twenty-fifth Anniversay Commemorative Issue.

[17] Lloyd Clifton Miller. 1995. *A study of form and content of Persian avaz.* Ph. D. Dissertation. University of Utah.

[18] Brendon Mizener and W Jay Dowling. 2022. Real-time modulation perception in Western Classical music. *Music Perception: An Interdisciplinary Journal* 39, 5 (2022), 484–502.

[19] M. Mohammadi. 2017. *Modal Modernities: Formations of Persian Classical Music and the Recording of a National Tradition.* Doctoral thesis 1 (Research UU / Graduation UU). Universiteit Utrecht.

[20] Magdalena Plewa and Bozena Kostek. 2015. Music Mood Visualization Using Self-Organizing Maps. *Archives of Acoustics* 40 (08 2015). doi:10.1515/aoa-2015-0051

[21] Max Reger. 2013. *Modulation.* Courier Corporation, 15 Wellman Avenue, North Chelmsford, MA 01863, USA.

[22] Felix Salzer. 1962. *Structural hearing: Tonal coherence in music.* Vol. 1. Courier Corporation, 15 Wellman Avenue, North Chelmsford, MA 01863, USA.

[23] Peter Thoegersen. 2024. *Maqam Melodies: Pitches, Patterns, and Developments of Music in the Middle East and other Microtonal Writings* (1 ed.). Jenny Stanford Publishing, Singapore. doi:10.1201/9781003492856

[24] George Tzanetakis, Manjinder Singh Benning, Steven R. Ness, Darren Minifie, and Nigel Livingston. 2009. Assistive music browsing using self-organizing maps. In *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments* (Corfu, Greece) *(PETRA '09)*. Association for Computing Machinery, New York, NY, USA, Article 3, 7 pages. doi:10.1145/1579114.1579117

[25] Reza Vali and Eric Barndollar. 2013. Arghonoon: A digital keyboard system for Persian music. https://www.cmu.edu/homepage/creativity/2013/spring/ensemble-idea.shtml Accessed: 2025-05-08.

[26] J. Vesanto and E. Alhoniemi. 2000. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks* 11, 3 (2000), 586–600. doi:10.1109/72.846731

[27] Giuseppe Vettigli. 2018. MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map. https://github.com/JustGlowing/minisom/ Version 2.3.3.