THE TWITTERING MACHINE

David Kim-Boyle
The University of Sydney
Sydney, Australia
david.kim-boyle@sydney.edu.au

ABSTRACT

This paper describes the development of *The Twittering Machine* (2021) for HoloLens 2 and prepared piano, which features a three-dimensional (3D) performance score holographically projected on the surface of the piano keyboard. The score presents a real-time visualization of Twitter tweets scraped during the performance and generated through the application of various Natural Language Processing (NLP) techniques. Various technical aspects of the work are discussed including the NLP processes, network architecture facilitating communication with the HoloLens 2, and techniques through which the holographic score is accurately mapped to the surface of the piano keyboard. The paper describes the work's aesthetic focus and details how mapping process from language to musical notation provides structural form.

1. INTRODUCTION

It is estimated that around 500 million tweets are posted to Twitter every day. From this extraordinary abundance of words much of which is fleetingly expressed and ephemeral in standing, Twitter has become not only a highly visible influence in political discourse [1] and protest [2], but also an invaluable source of data to help inform theorisation on a range of concepts such as trust [3], identity [4], and cultural appropriation [5]. Through its public Application Programming Interface (API), data scientists have even been able to explore how Twitter can provide important insights on human movement and mobility [6, 7] with instrumental application in fields ranging from economics to epidemiology.

For the author, Twitter presents first and foremost, an extraordinary map of differential relations which can be aestheticized in unique forms of musical expression. *The Twittering Machine*, for prepared piano and HoloLens 2 henceforth *TM*, explores this concept through a small microcosm of the Twittersphere, charting a sonic map of difference between tweets ostensibly similar in topic but often wildly divergent in expressed sentiment. The work features a 3D holographic score projected above the surface of the piano keyboard with features that constantly transform as incoming tweets are analyzed during the performance with various NLP techniques. Measure of difference or flux across tweets are determined and used to spatially transform (compress, rotate, stretch) this cartographic representation.

2. LANGUAGE

From the application of generative grammars [8] to facilitate understanding of the listening experience in the work of Lerdahl and Jackendoff [9] through to the exploration of Jakobson's theory of aphasics [10] in the work of composer Aaron Cassidy [11], linguistic insights and frameworks have been a source of inspiration for music theory, cognition, and creative practice for decades. More recently, with the development of powerful NLP libraries such as *spaCy*,[1] and *NLTK*,[2] which run in easy-to-use programming environments such as Python, unlike the intimidating development environments of early computational linguistic models [12], composers have unprecedented access to a wide range of powerful tools that can analyze a text's formal and semantic properties and provide data that may be applied to musical organization.

Exploring musically and aesthetically satisfying ways to create musical structure from NLP data was an important and particularly challenging stage in the development of *TM*. The poetic inspiration for the work, however, was not motivated by how these relationships might manifest. Rather, as previously noted, the work sought to focus on how qualities of difference and repetition propagate through the tweet-retweet paradigm that underpins much of the Twittersphere [13].

3. SCRAPING AND NLP

3.1 Twitter Scraping

The Twitter public API allows a plethora of data to be scraped from Twitter ranging from the textual content of a tweet and the number of likes or retweets that a tweet might have, through to tweets that have unique keywords or hashtags (#) embedded within their content. While the data types accessible through Twitter's API have ostensibly been guided by the needs of market and business analytics, privacy concerns have recently factored into these

[1] https://spacy.io

[2] https://nltk.org

determinations particularly with the 2021 deprecation of the ability to obtain the precise GPS location of scraped tweets, much to the chagrin of data analysts and others who may have used this information to provide helpful insights on social demographics [14].

In *TM*, tweets are scraped on a discrete keyword, e.g. "delta", with the Python library *Tweepy*,[3] every twenty seconds through the following generic API call –

```
for tweet in tweepy.Cursor(api.search, q=data, count=6,
lang="en", tweet_mode="extended").items();
```

where *data* corresponds to the stated keyword "delta". Additional conditionals are attached to the call to ensure only English language tweets are returned (using *lang="en"*) and that the entire text rather than a truncated text is returned (by using the command *tweet_mode="extended"*).

The subsequently returned text from each *Tweepy* call returns a string which is processed with the NLP Python libraries *spaCy* and *TextBlob*.[4]

3.2 Natural Language Processing

spaCy is a powerful Python library that returns a wealth of information on a text's formal structure and to a lesser depth, its semantic content. More advanced applications include the use of transformers for producing textual summaries, translations or developing chat bots. In *TM*, *spaCy* is applied at a relatively high-level to analyze a text's formal properties.

The *spaCy* pipeline first tokenizes a tweet by tagging each word with a part-of-speech (POS) tag which classifies its structural function within a sentence, as demonstrated in the simple example shown in Figure 1.

```
Autonomous cars shift insurance liability
towards manufacturers.

ADJ NOUN VERB NOUN NOUN ADP NOUN PUNCT
```

Figure 1. Sample POS tagging of a sentence. Text from *spaCy* documentation [15].

POS tagging is an important early stage of NLP more broadly as it facilitates dependency parsing, i.e. finding the relationships between the constituent words and phrases within a sentence. In *spaCy*, the first stage of dependency parsing, is to identify noun phrases, or *noun chunks* in the *spaCy* vernacular, within a sentence. Noun phrases are those chunks of a sentence which have a noun at the head. Curiously, in *spaCy* verb phrase identification is not part of the dependency parsing pipeline. Noun chunking of the sample sentence from Figure 1 returns the following –

```
chunk ['Autonomous', 'cars']
chunk ['insurance', 'liability']
chunk ['manufacturers']
```

Figure 2. Noun chunks of the sentence "Autonomous cars shift insurance liability towards manufacturers." Note that verbs ('shift'), adverbs, and adpositions ('towards') are not contained within the chunks.

In *TM* no further use of *spaCy* occurs beyond noun chunking however, to conclude the preceding example, once noun chunking has been completed, *spaCy* can produce a parse tree which presents the formal relationships between each constituent word of the sentence, see Figure 3.

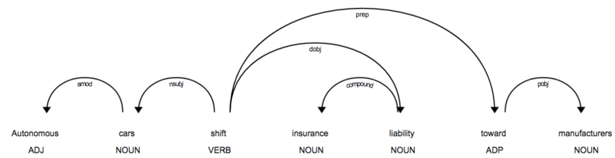

Figure 3. Graphic representation of the parse tree of "Autonomous cars shift insurance liability towards manufacturers." Example taken from *spaCy* documentation [15].

In *TM*, the very first tweet scraped upon commencement of a performance establishes a baseline against which all future tweets are measured for *similarity*. This measurement is undertaken with the *TextBlob* NLP Python library rather than *spaCy* as the latter can only provide similarity measurements between individual words rather than complete sentences and hence was deemed unsuitable. While *TextBlob* can also be used as a pipeline component within *spaCy*, in *TM*, it is used independently to measure the *similarity* between tweets but also the *sentiment* of an individual tweet.

In Twitter analytics, *sentiment* is expressed as a floating-point value ranging from -1.0 (negative sentiment) to +1.0 (positive sentiment) with tweets with *sentiment* values close to zero considered to be neutral. For example, a phrase such as "I love learning about the wonderful world of natural language processing" will return a more positive value for *sentiment* than "I hate studying natural language processing. It is very difficult." Similarity is a slightly more nuanced concept. While the two phrases just presented return polarized values for *sentiment*, they have similar subject matter. This is reflected in *TextBlobs* with a measurement known as a *similarity index*, a floating-point value ranging between 0.0 (highly dissimilar) and 1.0 (identical). The *similarity index* of the two preceding sentences is 0.83. In contrast, the *similarity index* of the first sentence of the pair with "All cows eat grass" is 0.12.

As an example of how NLP is applied in *TM*, which will be followed through in the next section, consider Figure 4 which presents two tweets returned on a scrape of keyword "delta."

[3] https://www.tweepy.org

[4] https://www.textblob.readthedocs.io/en/dev/

Figure 4. Two tweets containing keyword "delta" scraped on November 2nd, 2021.

A noun chunk analysis with *spaCy* of the first tweet of Figure 4 returns the following –

```
chunk ['Vaccine', 'impact', 'data']
chunk ['Vic']
chunk ['lower', 'hospitalization', 'burden']
chunk ['NSW', 'Delta', 'wave']
chunk ['higher', '💉', 'coverage']
```

Figure 5. The first five noun chunks from the first tweet of Figure 4.

A sentiment analysis with *TextBlob* returns the value 0.0625 suggesting a generally neutral tone with perhaps the phrase "Great to see…" skewing the result positively.

The returned *similarity index* between the two tweets of Figure 4 is 0.697, implying a degree of similarity that is ostensibly more apparent than real. While the two tweets are similar by the mere sharing of the keyword "delta" the semantic and contextual understanding is markedly different. Correcting for these nuances may be achieved by refining the search terms to include additional keywords or other flags that might yield more topically relevant results. The use of an additional keyword such as "vaccine" on the above search, for example, would not have returned the second tweet of Figure 4 and upon comparison with *TextBlob* would most likely returned a higher similarity measure.

4. VISUALIZATION AND MAPPING STRATEGIES

In *TM*, Twitter scraping and NLP is performed live in the Visual Studio IDE, and all associated data sent via a User Datagram Protocol (UDP) socket to a prototype Max patch. This patch was specifically developed to facilitate compositional planning and design from NLP data and tweet texts. While the processing of NLP data in Max could have been performed directly within Python, the ease with which intuitive graphical user interfaces (GUIs) can be built in the Max environment, together with its built-in tools for matrix analysis and transformation, was a particularly attractive feature during *TM*'s development.

In *TM* a tweet is visualized in the form of a set of colored nodes holographically projected above the surface of the piano keyboard via the HoloLens 2. A sample visualization of one such tweet is shown in Figure 6.

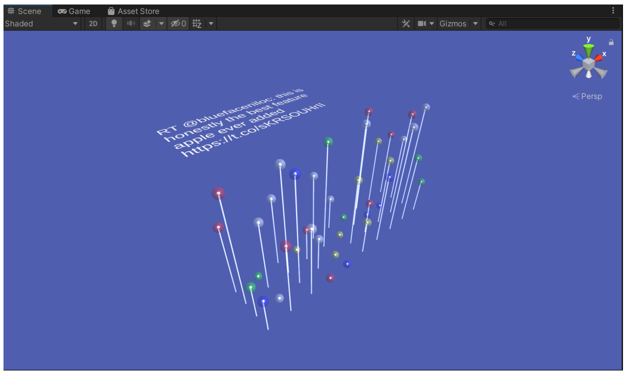

Figure 6. Visualization of a single tweet in *The Twittering Machine* from Unity3D's scene view. Note that the text of the tweet is included in the visualization. While this is mostly for visual aesthetics, it also helps demarcate node distribution groupings.

As can be seen from the above figure, there are only a small number of discrete graphic properties that constitute the visualization – node color, node size, node height, and node position with respect to pitch. From previous experience developing graphic scores, the constraints of our visual perception [16], and feedback from performers, the need to constrain the number of visual elements was an important factor for performability. In development of *TM*, then, there were two fundamental compositional questions to be asked, each of which was informed by the other, namely: 1) how might a tweet be visualized in the form of a performance score? and 2) how might that visualization be musically interpreted?

With respect to the first question, any visualization first requires an a priori decision about the type of data to visualize. In most cases this ultimately reduces to questions of utility or aesthetics. For *TM*, noun chunks, POS tags, tweet sentiment and similarity were found to yield the most useful and consistently usable material to help provide musical structure. At the same time, these linguistic features were most aligned with the aesthetic focus on difference and repetition.[5]

In *TM*, each word contained within a tweet text is represented by a node in the performance score. Node colors, with the exception of those nodes represented by the very first "baseline" tweet analyzed upon commencement of a

[5] In initial conceptualization, the use of precise geolocations embedded within tweets was of interest as this created the opportunity to draw correspondences between the geospatial propagation of tweets and retweets and proportional notational systems however, the deprecation of precise geotags from the Twitter API, meant that this data could no longer be gathered.

performance, are defined by noun chunks and POS tags. Each word within a noun chunk is assigned a uniform node color with colors in succeeding noun chunks cycling through colors red, green, blue, yellow, and orange, see Figure 7.

```
chunk ['Vaccine', 'impact', 'data'] -> [R, R, R]
chunk ['Vic'] -> [G]
chunk ['lower', 'hospitalization', 'burden'] -> [B, B, B]
chunk ['NSW', 'Delta', 'wave'] -> [Y, Y, Y]
chunk ['higher', '💉', 'coverage'] -> [O, O]
```

Figure 7. Mapping of words within noun chunks to node colors where ‘R’ = red, ‘G’ = green, ‘B’ = blue, ‘Y’ = yellow, and ‘O’ = orange. Note that emoticons, such as that represented in the fifth line, are not assigned a node in the performance score.

Any word not contained within a noun chunk (verbs, adverbs, adpositions etc.) which will have been identified by the POS tagging process, is represented in the performance score with a white-colored node. These nodes serve a different musical function in *TM* and thus need to be represented in a distinct way.

Musically, *TM* is built from harmonic structures (chords) and linear phrases. Each noun chunk is mapped to one of eight possible harmonic structures with each word within a chunk, or node in the performance score, assigned to a particular pitch within that structure, see Figure 8. Words that fall outside the noun chunks are mapped to a different set of pitches. The mapping and selection process is skewed by tweet *sentiment* and is managed within the prototype Max software.

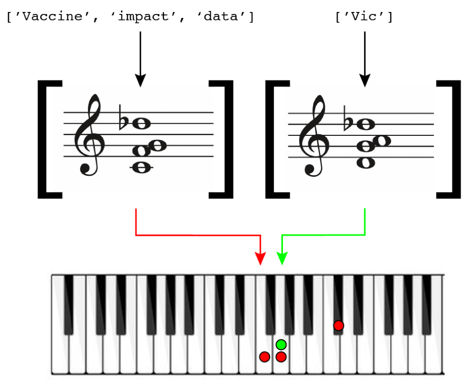

Figure 8. Sample mapping of noun chunk to harmonic structure to node positioning. The selection of active pitches from the harmonic structure is determined within the prototype Max software.

The harmonic structures of *TM* are bound within the interval of a major tenth. This constrains the range of pitches to ensure that tweets can be visualized to contiguous regions of the piano keyboard as shown in Figure 9. To facilitate visual recognition by the pianist, the tweet text from which NLP data is obtained is projected, with a left-justified margin, at the lowest note of the range, see Figure 6. Note that the first tweet visualized upon commencement of a performance, the initial baseline tweet against which all future tweets are compared, is anchored to the lowest region of the piano keyboard. As new NLP data is received via UDP, Max cycles through visualization mappings to adjacent keyboard regions.

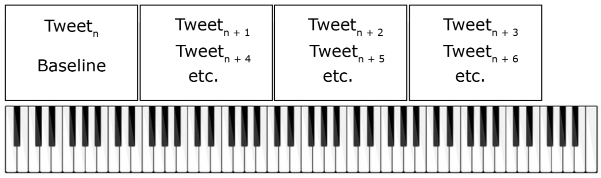

Figure 9. Pitch region mappings in *The Twittering Machine*.

The size of nodes in the performance score and the height above the keyboard at which they are visualized through the HoloLens 2, is also managed within the Max software and is not affected by incoming NLP data. Node height is interpreted by the pianist as an indication of duration while node size denotes dynamic level. Each of the nodes referenced by a word in a noun chunk is ascribed a uniform height and size. Nodes may be placed at a height of 50mm, 100mm, or 150mm above the piano keyboard surface corresponding to temporal durations of 5 seconds, 10 seconds, and 15 seconds respectively. Thin white lines are used to connect the key to the centre of nodes which helps facilitate node-pitch identification particularly when node density increases. White nodes are always positioned just above the surface of the keyboard at a height of 20mm. In a similar way to height assignments, nodes may be in one of three sizes, small (5mm), medium (10mm), large (20mm) corresponding to dynamic levels *pp*, *mp*, and *mf* respectively.

The first tweet analyzed upon commencement of a performance, the initial baseline tweet against which all future tweets are measured for similarity, is mapped and remains anchored to the lowest region of the piano keyboard. Unlike color mappings of all subsequent tweets, each word within this baseline tweet is mapped to a white colored node, again chosen from a predetermined set of pitches. As new tweet data is scraped and analyzed, the spatial distribution of the nodes contained within this baseline tweet visualization is transformed as measures of its similarity with new tweets varies. Spatial transformation of the visualization is performed with simple Max matrix rotation objects which displace node distributions within the octave. The two types of transformation are shown in Figure 10. The correlation of a spatial transform to a melodic permutation has, of course, numerous precedents in compositional practice most notably perhaps in the work of Xenakis [17] and Kagel [18] not to mention the ultimately spatial transformations of twelve-tone rows.

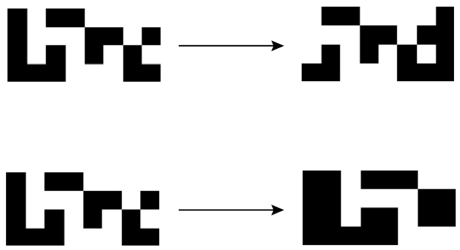

Figure 10. Matrix transformations – a) left displacement by one cell (upper), and b) stretching.

As new tweet information is visualized and transformations of the initial "baseline" tweet are processed, the interpretive options presented by the performance score vary. During performance, the pianist cycles through performance of each noun chunk harmonic unit in an order of their choosing. Linear phrases may be constructed from any visible white nodes or they may alternatively be performed as grace note filigrees to the harmonic structures referenced by noun chunks. The mapping of Figure 11a, for example, may be interpreted as shown in Figures 11b. For simplicity, dynamics and durations are not precisely prescribed.

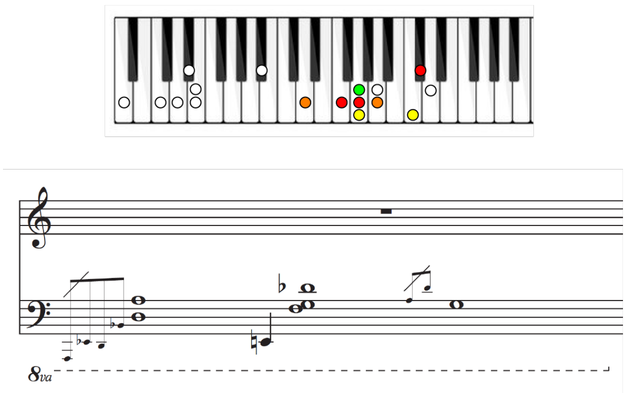

Figure 11. Possible interpretation of the performance score of *The Twittering Machine* – a) the performance score omitting node size and height information, and b) a possible interpretation.

5. TECHNICAL OVERVIEW

Although *TM* runs as a standalone application developed in Unity3D, it is dependent on both Twitter data, and the prototyping mapping software. Developed in C#, it comprises code that parses data received via UDP which is then mapped to the parameters of various 3D game objects (spheres, lines, and text fields). The developed application is then compiled deployed to the HoloLens 2 via the Visual Studio IDE.

5.1 Holographic Anchoring

The most significant technical challenge faced during development of the *TM* application, was the accurate spatial placement of the holograms. The work was initially developed on the original HoloLens hardware, a now deprecated device superseded by the HoloLens 2. Consequently, the readiest solution to the challenge of hologram placement was with fiducial markers, see Figure 12, placed on the inside lid of the piano. While there are several commercially available SDKs specifically designed to facilitate image detection and hologram placement, notably those developed by Vuforia,[6] and Wikitude,[7] neither proved to be effective solutions for *TM*. Vuforia exhibited significant latency between detection of an image and placement of a hologram and the reliability of image detection was inconsistent or at least not consistent enough for the purposes of live musical performance. The Wikitude SDK was even less suitable because it was unable to access the passthrough, built-in camera of the HoloLens.

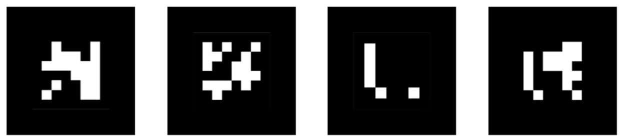

Figure 12. Sample markers used in hologram placements. These markers were placed at discrete locations inside the piano lid with hologram placement correlated accordingly, for example, displaced forward along the *z*-axis towards the pianist and slightly down along the *y*-axis towards the piano keyboard.

The most robust, reliable, and efficient image detection with the original HoloLens hardware was achieved with the ARToolkit through a modification developed by Long Qian [19] based on the use of ArUco markers [20].

On the HoloLens 2, many of the challenges experienced in accurate hologram anchoring had been resolved thanks to several new device affordances including built in QR-code detection,[8] and the extraordinarily powerful new object anchoring features integrated within Microsoft Azure. While the original HoloLens had the ability to use spatial anchors to facilitate the sharing of holograms across multiple users in the HoloLens 2, these were somewhat cumbersome to use and did not maintain persistence. In contrast, the HoloLens 2 features object anchoring, which unlike spatial anchors, can be aligned to objects such as a piano keyboard which will persist across instantiations. This capability mitigated the need to use fiducial markers physically attached to the piano, instead directly aligning holograms with particular piano keys which proved sufficiently reliable for application in *TM*.

[6] https://developer.vuforia.com
[7] https://www.wikitude.com
[8] https://docs.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview

5.2 UDP Control

In *TM*, the data received by the HoloLens 2 is sent from Max via the UDP protocol [21]. This requires the data to be processed as symbols and sent via standard *udpsend* objects to the internet protocol (IP) address of the HoloLens 2. For ease of development and testing, the four tweets visualized on the piano keyboard are assigned unique port numbers. As the data received by the HoloLens 2 via UDP consists of only score control data, issues of latency were not of concern.

Figure 13 presents a schematic of *The Twittering Machine*'s entire communication protocol.

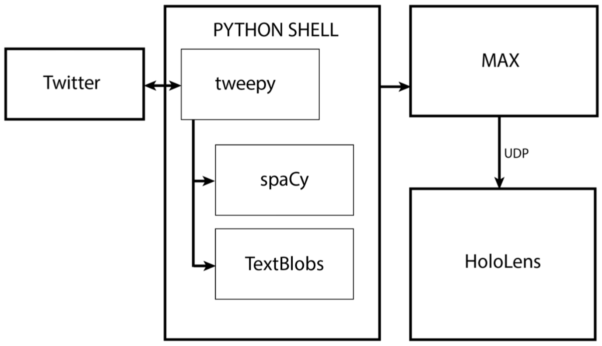

Figure 13. Technical schematic of *The Twittering Machine*.

6. FUTURE WORK

The development of *The Twittering Machine* involved the resolution of more complex technical challenges than any of my previous work with either the HoloLens or generative scores more broadly. While much of this was brought on by technical constraints and limitations, much was also related to challenges of a more aesthetic nature, namely on how Twitter data might be correlated to engaging musical structure and expression.

Further study into how the user experience (UX) of the pianist might impact score design considerations is also an area requiring considerable investigation. Factors such as head mounted display (HMD) comfort, color fidelity, and image stability [22], for example, are just some of the areas in which UX concerns play an important role in determining the effectiveness of a data visualization.

Extended reality (XR) hardware and applications are evolving rapidly with consumer awareness and interest in the metaverse ensuring a significant degree of concomitant commercial development. It is hoped that with the release of more consumer-focused hardware,[9] more researchers and creative practitioners will be able to leverage the affordances of the technology for innovative creative expression. The author is particularly interested in exploring how networking affordances might enable new forms of collaborative experience for users in a shared 3D virtual space and how these experiences might be facilitated through new representational paradigms.

[9] Consumer technology journals predict new mixed reality hardware from Apple, Facebook, and Magic Leap within the next 12-18 months.

7. REFERENCES

[1] J. H. Parmelee and S. L. Bichard, *Politics and the Twitter Revolution*. Lexington Books, 2011.

[2] A. R. Hosterman, N. R. Johnson, R. Stouffer, S. Herring, "Twitter, Social Support Messages, and the #MeToo Movement," in *The Journal of Social Media in Society*, vol. 7, no. 2, 2018, pp. 69-91. Available: https://www.thejsms.org/index.php/JSMS/article/view/475/198

[3] C. DeCarlo, *Twitter Culture: On the Trustworthiness of Tweets*. Enslow Publishing, 2018.

[4] A. Brock, "From the Blackhand Side: Twitter as a Cultural Conversation," in *Journal of Broadcasting & Electronic Media*, vol. 56, no. 4, 2012, pp. 529-549. Available: https://doi.org/10.1080/08838151.2012.732147

[5] D. Rosen (Ed.), *The Social Media Debate: Unpacking the Social, Psychological, and Cultural Effects of Social Media*, Routledge, 2022.

[6] R. Jurdak, K. Zhao, J. Liu, M. AbouJaude, M. Cameron, D. Newth, "Understanding Human Mobility from Twitter," *PLOS ONE*, vol. 10, no. 7, July, 2015. Available: https://doi.org/10.1371/journal.pone.0131469

[7] S. F. Khan, *Understanding Human Mobility Using Twitter*. PhD Thesis, The University of Queensland, 2020.

[8] N. Chomsky, *Syntactic Structures*. Mouton, 1957.

[9] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. MIT Press, 1983.

[10] R. Jakobson, *On Language*. 2nd Ed. Harvard University Press, 1995.

[11] A. Cassidy, *I, purples, spat blood, laugh of beautiful lips*. Self-published, 2006.

[12] L. Karttunen and K. R. Beesley, "Twenty-Five Years of Finite-State Morphology" in A. Arppe et al. (Eds.), *Inquiries into Words, Constraints and Contexts*, CSLI Publications, 2005.

[13] G. Deleuze, *Difference and Repetition*. Paul Patton (Trans.), Columbia University Press, 1995.

[14] A. Kruspe, M. Häberle, E. J. Hoffmann, S. Rode-Hasinger, K. Abdulahhad, and X. X. Zhu, "Changes in Twitter Geolocations: Insights and Suggestions for Future Usage," in *Proceedings of the 2021 EMNLP Workshop W-NUT: The Seventh Workshop on Noisy User-generated Text*, Association for Computational Linguistics, 2021, pp. 212-221.

[15] https://spacy.io

[16] W. Thompson, *Visual Perception from a Computer Graphics Perspective*. CRC Press, 2020.

[17] I. Xenakis, *Formalized Music*. Revised Ed. Pendragon Press, 1992.

[18] M. Kagel, “Translation/Rotation,” in *die Reihe*, vol. 7, 1965, pp. 32-60.

[19] L. Qian, A. Deguet, and P. Kazanzides, “ARssist: Augmented Reality on a Head-Mounted Display for the First Assistant in Robotic Surgery,” in *Healthcare Technology Letters*, vol. 5, no. 5, 2018, pp. 194-200.

[20] S. Garrido-Jurado, R. Muñoz-Salinas, F. José Madrid-Cuevas, and M. Jesús Marin-Jiménez, “Automatic Generation and Detection of Highly Reliable Fiducial Markers Under Occlusion,” *Pattern Recognition*, vol. 47, no. 6, 2014, pp. 2280-2292.

[21] D. Kim-Boyle and B. Carey, “3D Scores on the HoloLens,” in *International Conference on Technologies for Music Notation & Representation (TENOR 2019)*, Melbourne, 2019.

[22] D. Kim-Boyle, “96 Postcards in Real Color,” in *Proceedings of the TENOR2022 International Conference on Technologies for Music Notation and Representation*, Marseille, 2022.