# Automatically calculating tonal tension

Germán Ruiz-Marcos, Alistair Willis, and Robin Laney

The Open University
name.surname@open.ac.uk

**Abstract.** *Since the early years of the past century, many scholars have focused their efforts towards designing models to better understand the way listeners perceive musical tension. From the existing models, Lerdahl's has shown strong correlations against tension judgements provided by human listeners and has been used to make accurate predictions of musical tension. However, a full automation of Lerdahl's model of tension has not yet been made available. This paper presents a computational approach to automatically calculate musical tension according to Lerdahl's model, with a publicly available implementation.*

**Keywords:** Musical Tension; Automation; Lerdahl; TPS; GTTM.

## 1 Introduction

Music, in the Western tonal tradition, is often described by means of patterns of tension and relaxation (Krumhansl, 1996). However, automating the calculation of musical tension is not straightforward, mainly because many different psychoacoustic and cognitive features have shown a strong correlation with musical tension (see Granot and Eitan's (2011) literature review). In the last decade, there have been two notable attempts to automate the calculation of musical tension: Herremans and Chew (2016) and Farbood (2012). The former models the degree of tonal tension[1] of a given group of notes with regards to its dispersion in the tonal space, to its distance to the piece's key and to its movement in the "spiral array" (Chew's (2014) model of tonality). However, the authors noted that their model did not successfully consider features such as melodic contour and rhythm and would benefit from a more thorough empirical evaluation against listeners' judgements of perceived musical tension. The latter extracts musical trends from the behaviour of features such as dynamics, pitch height, tempo changes, onset frequency and harmonic motion. All these features but the last are well-defined in the literature and can be easily extracted from a piece of music automatically. However, Farbood's harmonic contribution depends on Lerdahl's (2004) model of tonal tension, of which an automatic implementation has not yet been made available.

Although several authors have already attempted to automate Lerdahl's work, the existing attempts show at least one of the following limitations. First,

[1] Henceforth, "tonal tension" specifically refers to the "sense [of tension] created by melodic and harmonic motion" (Lerdahl & Krumhansl, 2007, p.329).

some implementations do not consider Lerdahl's whole model of tension (e.g. De Haas, Wiering, and Veltkamp (2013) do not consider transitions across distant keys; Yoo and Lee (2006) do not consider the contribution a piece's structure might have in the calculation of tension). Second, some dimension in Lerdahl's model is not accurately implemented (e.g. Sakamoto, Arn, Matsubara, and Tojo (2016); Fukunari, Arn, and Tojo (2016); Matsubara, Ishiwa, Uehara, and Tojo (n.d.) do not consider Lerdahl's revised model concerning distant keys (Lerdahl & Krumhansl, 2007); Chang's (2013) implementation of the rules in Lerdahl's model, using pitch set theory, sometimes produces results which do not match the examples given in Lerdahl (2004)).

Motivated by these limitations, this paper aims at fully automating Lerdahl's model of tonal tension.

## 2 A Brief Review of Lerdahl's Model of Tonal Tension

In 1983, A Generative Theory of Tonal Music (GTTM) (Lerdahl & Jackendoff, 1983) was published, inspired by Schenkerian analysis and Chomky's linguistic theory. It was capable of modelling qualitative patterns of tension and relaxation through rule-based reductions of tonal pieces. Five years later, Lerdahl formalised a Tonal Pitch Space (TPS) (Lerdahl, 1988) to quantify the stability conditions which drive GTTM's rules. Comprising both GTTM and TPS, Lerdahl developed his Model of Tonal Tension (MTT) (Lerdahl, 2004), which has shown strong correlations against listeners' judgements of perceived tension (Lerdahl & Krumhansl, 2007) and has been used to make accurate predictions of musical tension (Farbood, 2012). However, it should be noted that this model of tension is limited to the repertoire of the Western tonal tradition.

GTTM establishes a collection of principles to define the most probable structures and note- and chord-relations a Westerner listener would infer within a piece of music. These principles are given as a system of rules that take into account the piece's patterns, its metrical structure and the hierarchical relations that might exist with regards to tonal stability. As a result, GTTM produces a hierarchical representation of the elements within the piece of music, in the form of a *tree*, which reflects qualitative patterns of tension and relaxation in the musical discourse.

TPS defines a *basic space* for any given chord. This space assigns a stability weighting to the twelve pitches in the chromatic scale according to the notes that shape the corresponding chord. The chord's root is given a value of 5, its fifth a value of 4, its third a value of 3, the rest of the pitches in the corresponding diatonic scale are given a value of 2 and the remaining non-diatonic pitches a value of 1. As a result, one can calculate the distance between two chords within TPS by comparing the number of steps needed to transform the *basic space* of the first chord into that of the second chord. By calculating the distances between the diatonic chords in every key, TPS can graphically represent the tonal system.

MTT provides a method for calculating a value of tension between two consecutive chords, known as *sequential tension*, $T_{seq}$. It depends on the distance between both chords within TPS and it also takes into account a collection of *surface parameters*. These parameters concern the scale degree of the chords' highest note, the chords' inversions and the role the chords' notes play in their respective *basic space*

To account for the impact a piece's hierarchical structure might have on the perception of tension, MTT includes a second method, known as *global tension*, $T_{glob}$. Its calculation is similar to that of *sequential tension*. However, in this case, it also includes inherited contributions to tension that are determined by the piece's *GTTM tree*.

To account for the tension produced by voice-leading relations, MTT includes a third and final method, known as *attraction*, $\alpha$. Its calculation compares updated versions of the chords' *basic spaces*, as well as the intervals between the voices within the chords transitions.

An example of the application of MTT on the Grail theme from Wagner's Parsifal is shown in Fig.1. It includes a *GTTM tree* and quantitative values of *global tension* and *attraction*. A detailed description of how these were calculated can be found in Lerdahl and Krumhansl (2007).

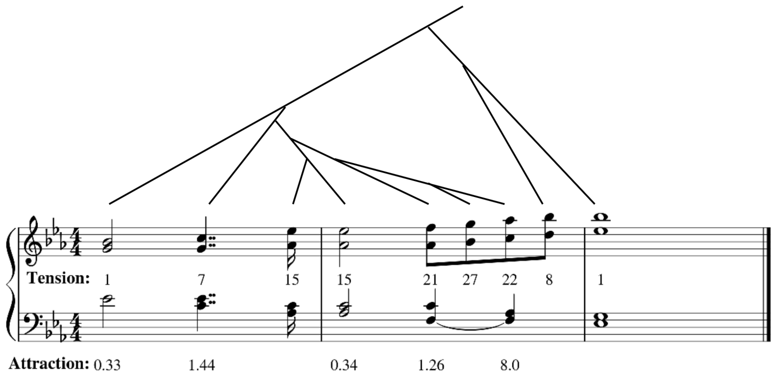


Fig. 1: MTT analysis performed on the Grail theme from Wagner's Parsifal.

A full discussion of GTTM, TPS and MTT is beyond the scope of this paper. The reader is referred to Lerdahl and Jackendoff (1983), Lerdahl (2004) and Lerdahl and Krumhansl (2007) for full details.

## 3 AuToTen: The Computational Approach

We here present AuToTen (as in **Au**tomatic **To**nal **Ten**sion), a publicly available repository[2] which stores a novel automation of Lerdahl's MTT.

[2] `https://doi.org/10.21954/ou.rd.13026578`

### 3.1 AuToTen's implementation

As input, AuToTen is fed with a piece of music and its GTTM representations, all in MusicXML format (Good et al., 2001) without additional notation. The latter need to be calculated, a priori, by the user. In this paper, the calculations of GTTM representations have been performed using the Interactive GTTM Analyser[3] (henceforth, IGA) (Hamanaka & Tojo, 2009), which is not hosted in AuToTen's repository nor maintained by this paper's authors. As output, AuToTen calculates the piece's quantitative values of *global tension* and *attraction*, as in Fig.1.

IGA re-formalises GTTM's rules using numerical expressions, with adjustable *control parameters*, to weight the priority of each rule. In this way, it is capable of automatically producing *GTTM trees* from a given piece of music. Note, however, that the accuracy of the generated trees will depend on the values at which the *control parameters* are set.

To calculate the values of *global tension* and *attraction* of a given piece of music, AuToTen performs the following steps:

**Step 1:** define the musical events that will be assigned with a tension value. To do so, AuToTen produces a list of the piece's *offsets* from the MusicXML input file which contains GTTM's metrical representation. These *offsets* are interpreted by AuToTen as the piece's beats which will be assigned with a value of *global tension* and *attraction* .

**Step 2:** represent the hierarchical relations in such a way that will facilitate the calculation of tension (recall that this representation, in the case of manual calculations, is given in the form of a *tree*, as in Fig.1). To do so, AuToTen produces a matrix (henceforth, *GTTM matrix*) from the MusicXML input file which contains GTTM's hierarchical representation.

**Step 3:** perform a harmonic analysis of the defined *offsets* and label them accordingly (Roman Numeral analysis is the notation used throughout Lerdahl's work). To do so, AuToTen calculates the piece's most suitable *key* and *chord* labels using the modules `analyze()` and `chordify()`, both built into the toolkit `music21` (Cuthbert & Ariza, 2010). See Tymoczko (2010) for more detail on the possibilities and accuracy of `music21`'s modules.

**Step 4:** calculate the *surface parameters* for all *offsets*. To do so, AuToTen applies MTT's rules to the *key* and *chord* associated which each *offset* using the values given by `music21`'s modules `getScaleDegreeFromPitch()`, `inversion()` and `pitches()`.

**Step 5:** calculate the values of *global tension* and *attraction* of all *offsets*. To do so, AuToTen is fed with the *surface parameters*, the *chords*' labels and the *GTTM matrix*, and produces the values of *global tension* and *attraction*, according to MTT's rules.

The above discussion is just a brief introduction to AuToTen's architecture. The reader is referred to AuToTen's repository for a complete description of its implementation.

[3] `http://www.gttm.jp/`

### 3.2 AuToTen's Evaluation

To evaluate the implementation of AuToTen, there are two questions that need to be answered. First, has Lerdahl's MTT been correctly implemented? And, second, how accurate are the calculations of tension provided by AuToTen?

To answer the first question, we have manually transcribed a total of 100 test cases and they can all be found in AuToTen's repository. From these, 50 cases concern TPS distances between chords: 30 typical close-chord transitions annotated from TPS' book (Lerdahl, 2004), 5 distant-chord transitions provided by F. Lerdahl (personal communication, September 27, 2019) and 15 atypical chord transitions that we manually calculated ourselves. The remaining 50 cases concern *global tension* values from two pieces of music (Lerdahl & Krumhansl, 2007): 9 events from Wagner's Grail theme from Parsifal and 41 events from Bach's chorale "Christus, der ist mein Leben". Once the test cases were carried out, the values of *surface parameters*, *global tension* and *TPS distance* calculated by AuToTen, for all 100 cases, agreed with the annotated ground-truths.

To answer the second question, four pieces of music have been used: the above Wagner's theme and Bach's chorale, and harmonic reductions of Chopin's E major prelude and the first phrase in Mozart's sonata k.282. The *GTTM trees* and chord labels, as well as the values of *global tension* and *attraction*, of these pieces can be found in Lerdahl and Krumhansl (2007); Lerdahl (2004). All these data were considered as ground-truth to evaluate AuToTen.

We have defined the chord labels' accuracy as the ratio of labels that matched those in the ground-truth. These accuracy values are shown in the first two rows in Table 1. Notice that we have considered two scenarios. One scenario refers to the differences between the estimated chord labels when only considering *diatonic chords* in **Step 3**, and another scenario where *non-diatonic chords* are also considered by applying `music21`'s `chordify()` to every measure's *key*.

Table 1: Chord labels' and GTTM trees' accuracy values.

| ***Accuracy measurement*** | | ***Wagner's piece*** | ***Mozart's piece*** | ***Bach's piece*** | ***Chopin's piece*** |
|---|---|---|---|---|---|
| ***Chord labels'*** | *just diatonic* | .89 | .62 | .63 | .44 |
| | *also non-diatonic* | .89 | .69 | .71 | .53 |
| ***GTTM trees'*** | *default GTTM tree* | .22 | .08 | .17 | .04 |
| | *edit GTTM tree* | .56 | .39 | .38 | .04 |

Symbolic chord recognition is still a challenging task and its performance accuracy heavily depends on the input music being used (Delgado & Nápoles, 2017). The chord labels' accuracy shown in Table 1 average to .65, in the case of *just diatonic chords*, and .71 when incorporating *non-diatonic chords*. We believe both values set a good starting point for this first version of AuToTen. Because of the increment of the accuracy from the *diatonic* to the *non-diatonic* scenario, in all four pieces of music, we have set the *non-diatonic* algorithm as the final version implemented in AuToTen. On the other hand, these values should not be taken as an indicator of the capabilities of `music21` nor AuToTen. Notice that GTTM's input should be homophonic music which eases the chord recognition task. Likewise, we believe most chord labels' mismatchings are due to two main reasons. First, sometimes `music21` interprets ornaments (e.g. passing notes or flourishes) as chords. Second, although `music21`'s `analyze()` has been applied to calculate every measure's key, and therefore identify non-diatonic chords, it seems that it still does not capture heavy modulation processes. For example, we believe the chords' accuracy value of .53 in Chopin's piece, as shown in Table 1, is due to the prelude's last two phrases transitioning across seven keys. Because of this, we believe future versions of AuToTen should explore `music21`'s `derivation` module to check whether a chord is derived from an element in a different key.

We have defined the *GTTM tree*'s accuracy as the ratio of rows in the *GTTM matrix* that matched those in the ground-truth's *matrix*. These accuracy values are shown in the last two rows in Table 1. Notice that we have also considered two scenarios. One scenario refers to the differences between the estimated *GTTM trees* when using the *control parameters* set by *default* in IGA's interface. In the other scenario, up to four of these *control parameters* were manually edited to get more accurate *GTTM trees*.

As seen in Table 1, there is an increment of the *GTTM trees*' accuracy in all pieces when editing the *default control parameters* in IGA, other than Chopin's, which remains the same. Still, the average *edit* accuracy just reaches .33. Nevertheless, two issues should be noticed here. First, the matrix accuracy relates to conditional probabilities, as the probability of a tree branch being right-labelled depends on the probabilities of the branches above it. A further analysis of this conditioning should be addressed in future research, maybe examining the salience of the trees' branches, as proposed in Marsden, Tojo, and Hirata (2018). Second, note that the *edit* scenario only included up to four modifications of IGA *control parameters*. In Hamanaka, Isono, Hirata, and Tojo (2020), more than 30 modifications, on average, were needed in an IGA-based user study. What is more, the authors stressed how time-consuming and difficult using IGA could be. For the sake of simplicity, we decided to apply very little modifications to IGA's *control parameters*. It seems that these modifications already translate into greater values of accuracy, but there is still room for further research on deeper IGA-based analysis. Likewise, a more accessible GTTM editor was recently presented in Hamanaka et al. (2020), which could be used by the user, instead of IGA, to calculate the *GTTM trees*. Because of the increment of the accuracy from the *default* to the *edit* scenario, the results shown later in the paper will always refer to *GTTM trees* calculated in the *edit* scenario.

Lerdahl and Krumhansl (2007) tested MTT against the tension perceived by human listeners using the pieces in Table 1 except Mozart's. They calculated a multiple regression using the listeners' judgements, $j$, as the dependent variable and MTT's *global tension*, $T_{glob}$, and *attraction*, $\alpha$, as the independent variables. That means they calculated $P = \beta_t \cdot T_{glob} + \beta_\alpha \cdot \alpha$ and its $R^2(n, m)$, where the betas denote the regression's coefficients, $R^2$ is the proportion of variation in the data accounted by MTT, $(n, m)$ represent the degrees of freedom ($n$: number of independent variables; $m$: number of data points minus two), and $P$ denotes MTT's predictions that best approximate to $j$. Only if $R^2 = 1$ does $j = P$. Fig.2 (taken from Lerdahl and Krumhansl (2007)) shows the average tension perceived by human listeners, $j$, when listening to Wagner's piece against the predictions of tension calculated by MTT, $P$, according to the values in Fig.1.

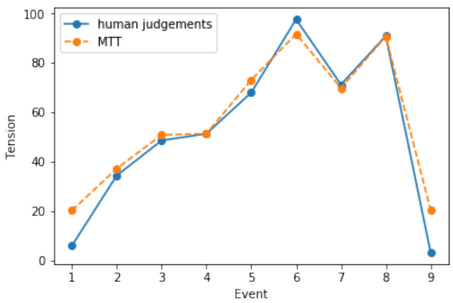


Fig. 2: MTT vs human judgements.

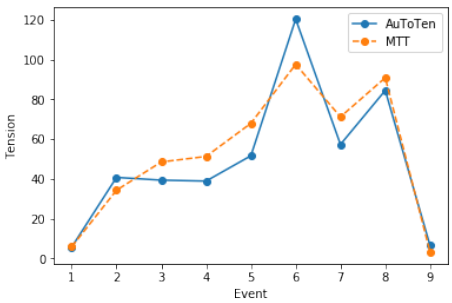


Fig. 3: AuToTen vs MTT.

To test AuToTen's accuracy, we have analysed its outputs correlation against MTT's $T_{glob}$, $\alpha$ and $P$, as shown in Table 2. In the case of Mozart's sonata, $P$ was not included in its empirical study (Krumhansl, 1996). We calculated this ourselves using the corresponding data provided by C. Krumhansl (personal communication, October 09, 2019).

Table 2: Results of the regressions of the values of $T_{glob}$ (first row), $\alpha$ (second row) and $P$ (third row) calculated by AuToTen against those in Lerdahl and Krumhansl (2007); Lerdahl (2004).

| | **Wagner's piece** | **Mozart's piece** | **Bach's piece** | **Chopin's piece** |
|---|---|---|---|---|
| ***MTT's $T_{glob}$ against AuToTen's $T_{glob}$*** | $R^2(1,7) = .8$<br>$p_t = .0005$ | $R^2(1,11) = .75$<br>$p_t < .0001$ | $R^2(1,39) = .72$<br>$p_t < .0001$ | $R^2(1,45) = .61$<br>$p_t < .0001$ |
| ***MTT's $\alpha$ against AuToTen's $\alpha$*** | $R^2(1,7) = .92$<br>$p_\alpha < .0001$ | $R^2(1,11) = .91$<br>$p_\alpha < .0001$ | $R^2(1,39) = .73$<br>$p_\alpha < .0001$ | $R^2(1,45) = .71$<br>$p_\alpha < .0001$ |
| ***MTT's $P$ against AuToTen's $T_{glob}$ and $\alpha$*** | $R^2(2,7) = .95$<br>$p_t < .0001$<br>$p_\alpha = .014$ | $R^2(2,11) = .83$<br>$p_t = .002$<br>$p_\alpha = .015$ | $R^2(2,39) = .8$<br>$p_t < .0001$<br>$p_\alpha = .005$ | $R^2(2,45) = .67$<br>$p_t < .0001$<br>$p_\alpha = .008$ |

As it can be seen in Table 2, both *global tension*, $T_{glob}$, and *attraction*, $\alpha$, show a strong correlation for all four pieces of music ($R_t^2 = .72$; $R_\alpha^2 = .82$, on average), with the exception of Chopin's $T_{glob}$ regression, whose correlation is moderately strong (or "healthy", a term used by Lerdahl and Krumhansl (2007, p.349) to refer to this range of values). Likewise, following convention ("$R^2$ is considered significant when the $p$ value is less than .05" (Lerdahl & Krumhansl, 2007, p.341)), they are both statistically significant, again for all four pieces of music ($p_t \leq .0005$; $p_\alpha < .0001$). The regressions of both $T_{glob}$ and $\alpha$ against MTT's predictions, $P$, show again a strong correlation in the case of all pieces but that of Chopin, whose correlation is healthy, and are all statistically significant ($R^2 = .81$ on average; $p_t \leq .002$; $p_\alpha \leq .015$). Notice how, as the number of events increases (Bach and Chopin), the correlation tends to decrease because there are more possible points of deviation (see the second degree of freedom). Fig.3 illustrates the regression data of the last row in Table 2 concerning Wagner.

One can calculate an adjusted value, $R^2_{adj}$, of $R^2$ to make it "more comparable with other models for the same data that have different numbers of degrees of freedom" (Lerdahl & Krumhansl, 2007, p.341). The average $R^2_{adj}$ of MTT's correlation against listeners judgements, calculated from Lerdahl and Krumhansl data, equals .72. The average $R^2_{adj}$ of AuToTen's correlation against MTT's predictions equals .8.

## 4 Conclusions

This paper has presented AuToTen, a system able to calculate the values of tension of a given piece of tonal music according to Lerdahl's model of tonal tension (MTT). In developing AuToTen, we have demonstrated how to "tighten [Lerdahl's] entire theory to the point that is implemented computationally", as Lerdahl proposed as a future prospect more than a decade ago (Lerdahl, 2009, p.193). Likewise, we have provided an architecture whose components can be independently used or further developed by the community.

In future, we intend to produce a more detailed report of AuToTen's implementation. We believe the community will benefit from a thorough discussion about the computational challenges we encountered when automating MTT, and how we overcome them. Likewise, several directions of future work have been considered. These include exploration of Hamanaka et al.'s (2020) new GTTM editor and the incorporation of `music21`'s `derivation` module in AuToTen's implementation.

To sum up, we believe that AuToTen can have a great impact in the community. For instance, it can be used to produce datasets of tonal tension, currently lacking from the literature (Herremans, Chuan, & Chew, 2017; Mizutani & Iwami, 2017); it can enable automating other models of musical tension, such as that of Farbood (2012), whose only missing automatic feature is Lerdahl's model; it can also foster the automatic generation of music matching tension, as in Herremans and Chew (2017); and has many more applications.

**Acknowledgements** We would like to thank F. Lerdahl and C. Krumhansl for openly discussing some of MTT's theoretical and methodological concepts with us, as well as for providing the data needed to test AuToTen.

## References


Chang, S. (2013). *Modeling tonal tension in music signals* (Unpublished doctoral dissertation). Graduate School of Convergence Science and Technology at Seoul National University.

Chew, E. (2014). *Mathematical and computational modeling of tonality: Theory and applications* (Vol. 204). doi: 10.1007/978-1-4614-9475-1

Cuthbert, M. S., & Ariza, C. (2010). music21: A toolkit for computer-aided musicology and symbolic music data.

De Haas, W. B., Wiering, F., & Veltkamp, R. C. (2013). A geometrical distance measure for determining the similarity of musical harmony. *International Journal of Multimedia Information Retrieval*, *2*(3), 189–202.

Delgado, N., & Nápoles, N. (2017, April 04). *Symbolic chord detection.* Music Information Retrieval Student blog for the MIR course, Master in Sound and Music Computing, Universitat Pompeu Fabra (2017). Retrieved from `https://musicinformationretrieval.files.wordpress.com/2017/04/symbolic-chord-detection.pdf`

Farbood, M. M. (2012). A parametric, temporal model of musical tension. *Music Perception: An Interdisciplinary Journal*, *29*(4), 387–428.

Fukunari, T., Arn, S., & Tojo, S. (2016). Ccg analyzer with tonal pitch space for non-classical chords. In *2016 eighth international conference on knowledge and systems engineering (kse)* (pp. 239–246).

Good, M., et al. (2001). Musicxml: An internet-friendly format for sheet music. In *Xml conference and expo* (pp. 03–04).

Granot, R. Y., & Eitan, Z. (2011). Musical tension and the interaction of dynamic auditory parameters. *Music Perception: An Interdisciplinary Journal*, *28*(3), 219–246.

Hamanaka, M., Isono, Y., Hirata, K., & Tojo, S. (2020). Web-based time-span tree editor and analysis database. In Proceedings of the 17th Sound and Music Computing Conference, Torino, June 24th – 26th 2020.

Hamanaka, M., & Tojo, S. (2009). Interactive gttm analyzer. In *Ismir* (pp. 291–296).

Herremans, D., & Chew, E. (2016). Tension ribbons: Quantifying and visualising tonal tension.

Herremans, D., & Chew, E. (2017). Morpheus: generating structured music with constrained patterns and tension. *IEEE Transactions on Affective Computing*.

Herremans, D., Chuan, C.-H., & Chew, E. (2017). A functional taxonomy of music generation systems. *ACM Computing Surveys (CSUR)*, *50*(5), 1–30.

Krumhansl, C. L. (1996). A perceptual analysis of mozart's piano sonata k. 282: Segmentation, tension, and musical ideas. *Music Perception: An Interdisciplinary Journal*, *13*(3), 401–432.

Lerdahl, F. (1988). Tonal pitch space. *Music Perception: An Interdisciplinary Journal*, *5*(3), 315–349.

Lerdahl, F. (2004). *Tonal pitch space.* Oxford University Press.

Lerdahl, F. (2009). Genesis and architecture of the gttm project. *Music Perception: An Interdisciplinary Journal*, *26*(3), 187–194.

Lerdahl, F., & Jackendoff, R. S. (1983). *A generative theory of tonal music.* MIT press.

Lerdahl, F., & Krumhansl, C. L. (2007). Modeling tonal tension. *Music Perception: An Interdisciplinary Journal*, *24*(4), 329–366.

Marsden, A., Tojo, S., & Hirata, K. (2018). No longer'somewhat arbitrary' calculating salience in gttm-style reduction. In *Proceedings of the 5th international conference on digital libraries for musicology* (pp. 26–33).

Matsubara, M., Ishiwa, Y., Uehara, Y., & Tojo, S. (n.d.). Computational detection of local cadence on revised tps.

Mizutani, T., & Iwami, Y. (2017). Analysis of musical expression using tonal tension and attraction models. *DEStech Transactions on Engineering and Technology Research*(amsm).

Sakamoto, S., Arn, S., Matsubara, M., & Tojo, S. (2016). Harmonic analysis based on tonal pitch space. In *2016 eighth international conference on knowledge and systems engineering (kse)* (p. 230-233).

Tymoczko, D. (2010). *A geometry of music: Harmony and counterpoint in the extended common practice.* Oxford University Press.

Yoo, M.-J., & Lee, I.-K. (2006). Musical tension curves and its applications. In *Icmc.*