

# The Melody Triangle - Pattern and Predictability in Music

Henrik Ekeus (1), Samer Abdallah (1),  
Mark D. Plumbley, Peter W. McOwan  
(1) Centre for Digital Music  
Queen Mary University of London

## ABSTRACT

The Melody Triangle is a Markov-chain based melody generator where the input - positions within a triangle - directly map to information theoretic measures of its output. The measures are the entropy rate, redundancy and *predictive information rate*[?] in the melody. Predictive information rate is a *time-varying* information measure developed as part of the Information Dynamics of Music project (IDyOM)<sup>1</sup>. It characterises temporal structure and is a way of modelling expectation and surprise in the perception of music.

We describe the Information Dynamics model and how it forms the basis of the Melody Triangle. We outline two interfaces and uses of the system. The first is a multi-user installation where collaboration in a performative setting provides a playful yet informative way to explore expectation and surprise in music. The second is a screen based interface where the Melody Triangle becomes a compositional tool for the generation of intricate musical textures using an abstract, high-level description of predictability. Finally we outline a study where the screen-based interface was used under experimental conditions to determine how the three measures of predictive information rate, entropy and redundancy might relate to musical preference. We found that...

## Keywords

Information dynamics, Markov chains, Collaborative performance, Aleatoric composition

## 1. INFORMATION DYNAMICS

Music involves patterns in time. When listening to music we continually build and re-evaluate expectations of what is to come next. Composers commonly, consciously or not, play with this expectation by setting up expectations which may, or may not be fulfilled. This manipulation of expectation and surprise in the listener has been articulated by [ref]. [little more background on expectation]

<sup>1</sup><http://www.idyom.org/>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NIME12, Somewhere

Copyright 2008 Copyright remains with the author(s).

The research into Information Dynamics explores several different kinds of predictability in musical patterns, how human listeners might perceive these, and how they shape or affect the listening experience. [more on IDyOM project]

## 2. THE MELODY TRIANGLE

The Melody Triangle is based on first order Markov chains, represented as transition matrixes, that generate streams of symbols. By mapping the symbols to individual notes, melodies are generated. Further by layering these streams of notes can result in intricate musical textures. The choice of notes or scale is not a part of the Melody Triangle's core functionality, in fact the symbols could be mapped to anything, even non sonic outputs.

Any sequence of symbols can be analysed and information theoretic measures taken from it. The novelty of the Melody Triangle lies in that we go 'backwards' - given desired values for these measures, as determined from the user interface, we return a stream of symbols that match those measures. The information measures used are redundancy, entropy rate and predictive information rate.

### 2.1 Information measures

#### 2.1.1 Redundancy

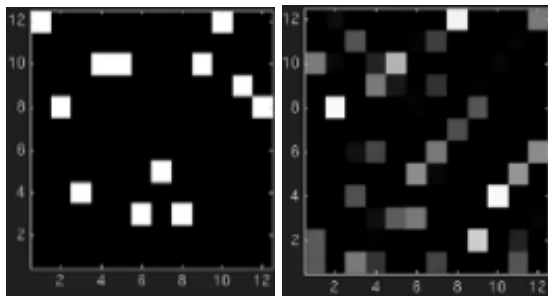
[todo - a more formal description] Redundancy tells us the difference in uncertainty before we look at the context (the fixed point distribution) and the uncertainty after we look at context. For instance a transition matrix with high redundancy, such as one that represents a long periodic sequence, would have high uncertainty before we look at the context but as soon as we look at the previous symbol, the uncertainty drops to zero because we now know what is coming next.

#### 2.1.2 Entropy rate

[todo - a more formal description] Entropy rate is the average uncertainty for the next symbol as we go through the sequence. A looping sequence has 0 entropy, a sequence that is difficult to predict has high entropy rate. Entropy rate is an average of surprisingness over time.

#### 2.1.3 Predictive Information Rate

[todo - a more formal description] Predictive information rate tell us the average reduction in uncertainty upon perceiving a symbol; a system with high predictive information rate means that each symbol tells you more about the next one. If we imagine a purely periodic sequence, each symbol



**Figure 1:** Two transition matrixes. The shade of white represents the probabilities of transition from one symbol to the next (black=0, white=1). The current symbol is along the bottom, and in this case there are twelve possibilities (mapped to a chromatic scale). The left hand matrix has no uncertainty; it represents a periodic pattern. We can see for example that symbol 4 must follow symbol 3, and that symbol 10 must follow symbol 4, and so on. The right hand matrix contains unpredictability but nonetheless is not completely without perceivable structure, it is of a higher entropy rate.

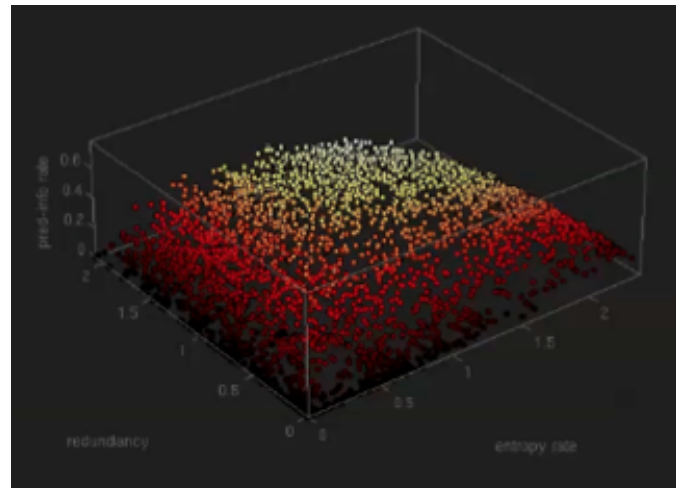
tells you nothing about the next one that we didn't already know as we already know how the pattern is going. Similarly with a seemingly uncorrelated sequence, seeing the next symbol does not tell us anymore because they are completely independent anyway; there is no pattern. There is a subset of transition matrixes that have high predictive information rate, and it is neither the periodic ones, nor the completely un-correlated ones. Rather they tend to yield output that have certain characteristic patterns, however a listener can't necessarily know when they occur. However a certain sequence of symbols might tell us about which one of the characteristics patterns will show up next. Each symbols tell a us little bit about the future but nothing about the infinite future, we only learn about that as time goes on; there is continual building of prediction.

## 2.2 Making the triangle

We generate hundreds of transition matrixes, representing first-order Markov chains, by a random sampling method. These are then plotted in a 3d statistical space of redundancy, entropy rate and predictive information rate. In figure 2 we see a representation of how these matrixes are distributed; each one of these points corresponds to a transition matrix.

When we look at the distribution of randomly generated transition matrixes and plotted in this space, we see that it forms an arch shape that is fairly thin. It thus becomes a reasonable approximation to pretend that it is just a sheet in two dimensions; and so we stretch out this curved arc into a flat triangle. It is this triangular sheet that is our 'Melody Triangle' and forms the interface by which the system is controlled.

When the Melody Triangle is used, regardless of whether it is as a screen based system, or as an interactive installation, it involves a mapping to this statistical space. As can be seen in figure 3, a position within the triangle maps to different measures of redundancy, entropy rate and predictive information rate.



**Figure 2:** The population of transition matrixes distributed along three axes of redundancy, entropy rate and predictive -information rate. Note how the distribution makes triangle-like plane floating in 3d space.

[todo improve, include example melodies?] Each corner corresponds to three different extremes of predictability and unpredictability, which could be loosely characterised as periodicity, noise and repetition. Melodies from the 'noise' corner have no discernible pattern; they have high entropy rate, low predictive information rate and low redundancy. These melodies are essentially totally random. A melody along the 'periodicity' to 'repetition' edge are all deterministic loops that get shorter as we approach the 'repetition' corner, until it becomes just one repeating note. It is the areas in between that provide the more interesting melodies, those that have some level of unpredictability, but are not completely random and conversely that are predictable, but not entirely so. This triangular space allows for an intuitive exploration of expectation and surprise in temporal sequences.

## 3. USER INTERFACES

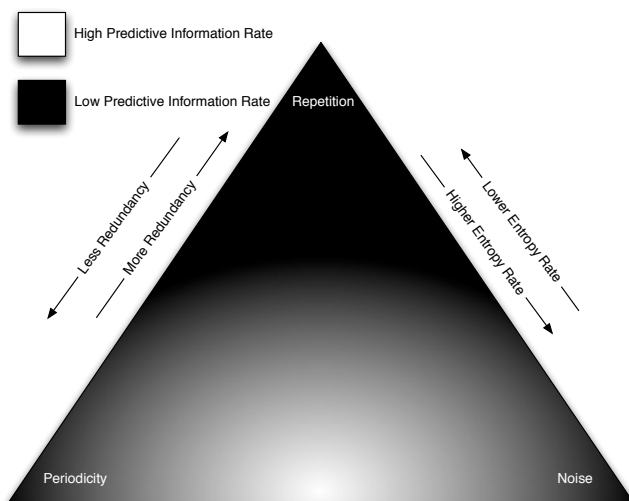
The Melody Triangle engine, developed in Prolog and MatLab, can be controlled by OSC messages and thus any number of interfaces could be developed to for it. Currently two different interfaces exist; a standard screen based interface where a user moves tokens with a mouse in and around a triangle on screen, and a multi-user interactive installation where a Kinect camera tracks individuals in a space and maps their positions in the space to the triangle.

### 3.1 The Multi-User Installation

[old text - rewrite!]

the statistical properties of this melody is based on where in the physical room the participant is standing, as this is mapped to a statistical space (see below). By exploring the physical space the participants thus explore the predictability of the melodic and rhythmical patterns, based on a simple model of how might guess the next musical event given the previous one. ...

When multiple people are in the space, they can cooperate to create musical polyphonic textures. For example, one



**Figure 3: The Melody Triangle** [todo fix shading to be more triangular]

person could lay down a predictable repeating bass line by keeping themselves to the periodicity/repetition side of the room, while a companion can generate a freer melodic line by being nearer the 'noise' part of the space.

A Kinect camera was used to track individuals in the space, an application developed in OpenFrameworks would send the individuals positions and a bounding box values (for gesture recognition), to an application running the Information Dynamics Engine [Matlab/Prolog]. This would map the input to a series of symbols that would go over OSC to Max/MSP, where these would be interpreted as MIDI, mapped to channels and the data passed on to Logic for output using software instruments.

### 3.1.1 Tracking and Control

It uses the OpenNI libraries' API and high level middleware for tracking with Kinect. This provided reliable blob tracking of humanoid forms in 2d space. By then triangulating this to the Kinect's depth map it became possible to get reliable coordinate of visitors positions in the space.

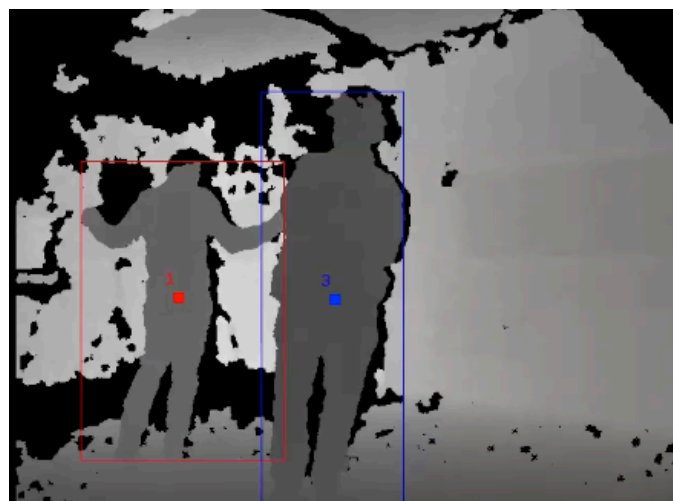
This system was then further extended to detect gestures. By detecting the bounding box of the 2d blobs of individuals in the space, and then normalising these based on the distance of the depth map it became possible to work out if an individual had a an arm stretched out or if they were crouching.

With this it was possible to define a series of gestures for controlling the system without the use of any controllers. Thus for instance by sticking out one's left arm quickly, the melody doubles in tempo. By pulling one's left arm in at the same time as sticking the right arm out the melody would shift onto the offbeat. Sending out both arms would change instrument.

## 3.2 The Screen Based Interface

[todo] [screen shot] On the screen is a triangle and a round token.

With the mouse you can click and drag the red token and move it around the screen. When the red token is dragged



**Figure 4: The depth map as seen by the Kinect, and the bounding box outlines the blobs detected by OpenNI.**

left arm	right arm	meaning
out	static	double tempo
in	static	halve tempo
static	out	triple tempo
static	in	one-third tempo
out	in	shift to off-beat
out	out	change instrument
in	in	reset tempo

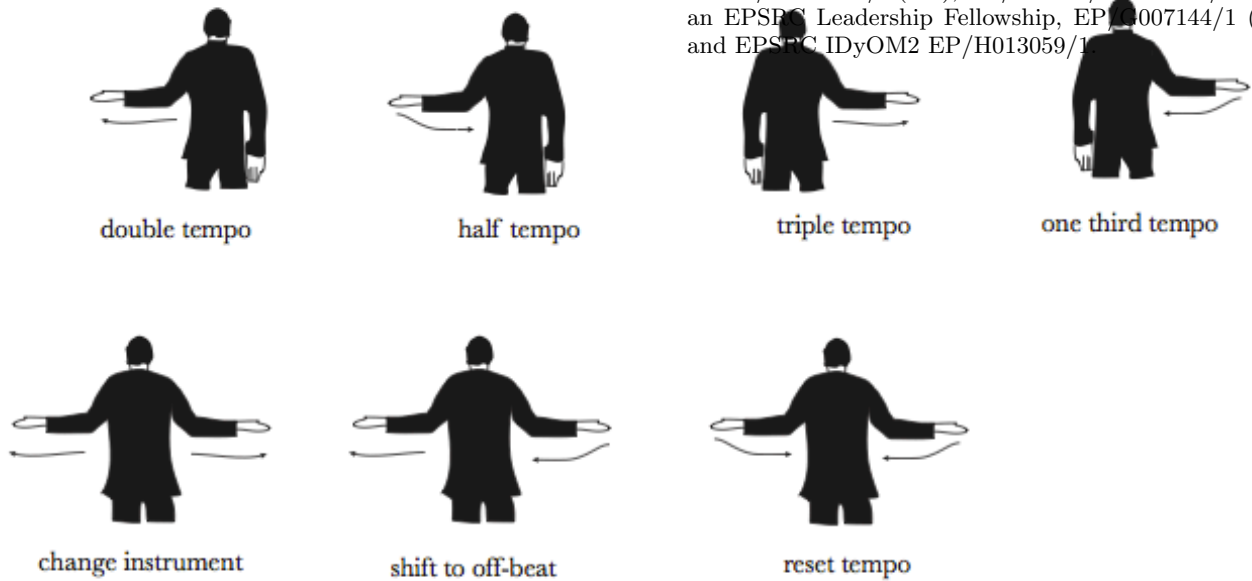


Figure 5: Control Gestures

into the triangle, the system will start generating a sequence of piano notes. The pattern of notes depends on where in the triangle the token is

#### 4. MUSICAL PREFERENCE AND INFORMATION DYNAMICS STUDY

[TODO!] The study was divided in to 5 subtasks. The axes of the triangle would be randomly rearranged prior for each participant.

this first task, which will last [4/3] minutes, we simply ask you to move the token where ever in the triangle you wish,. This allowed the participant to get use to the environment get use to the interface and get a sense of how position of tokens changes a melody.

In the following tasks a background melody is playing and the participant are asked to find a second melody that 'works well' with the background melody. In each of these tasks the background melody has different statistical properties. In the first it ....., In the second the background melody ... in the third... And finally in the fourth case the melody is in the middle of the triangle, that is it....

##### 4.1 Results

X participants took part in the study (mean median age). (Prior musical experience? )

##### 4.2 Observation/Discussion

#### 5. FURTHER WORK

#### 6. ACKNOWLEDGMENTS