

“To some extent, this match is a defense of the whole human race. Computers play such a huge role in society. They are everywhere. But there is a frontier that they must not cross. They must not cross into the area of human creativity. It would threaten the existence of human control in such areas as art, literature, and music.”

-Garry Kasparov in 1996, a year before being beaten by chess program, Deep Blue



Flatt, Scruggs and Markov



Goal: Simulate the sound of a bluegrass band with Markov generated fiddle lines, algorithmically generated banjo harmony, and user selected recombinatorial drums, bass, and guitar accompaniments.

“Music is the pleasure the human soul experiences from counting without being aware that it is counting.”

-Gottfried Leibniz

User

The user could be someone who enjoys bluegrass music, someone interested in computer generated composition, someone interested in computational creativity, or someone looking for entertainment.

Input 1

Musical Phrases



Markov Matrix Seed Phrases

Mike Cirillo

(A) tonic phrase



3 (b min) supertonic phrase



5 (c# min) mediant phrase



7 (D) subdominant phrase



9 (E7) dominant phrase



11 (f# min) submediant phrase



13 (G) flat 7 phrase



Input 2

User Input:

1. Chord Progression
2. Instruments in mix
3. Tempo
4. Markov depth

Structure

The macrostructure comes from the user selected broad tonal areas, all bookended within a traditional bluegrass musical introduction and ending riff.

The microstructure within the tonal areas comes from the composed Markov matrices.

Meaning

The target meaning is that of a bluegrass band.

Any additional associations (The Beverly Hillbillies, Deliverance) depends on the user.

Evoking Meaning:

1. sound fonts

I use sound fonts to evoke the impression of fiddle, banjo, guitar, bass, and drums. This is done through the sound shape and timbre.

2. recombinatorial composed fragments

I compose the sounds in ways indicative of a bluegrass band. I use authentic guitar voicing, and offset the attack of each note to simulate physical strumming.

3. “fiddle” seed compositions

I compose the Markov matrices to generate phrases indicative of bluegrass fiddle.

4. “banjo” harmony algorithms

I compose the harmony logic to be consistent with what a banjo player might actually play.



Demo

“We are incredibly attuned to the idea that the sole purpose of our technology is to solve problems. It also creates concepts and philosophy. We must more fully explore these aspects of our inventions, because the next generation of technology will speak to us, understand us, and perceive our behavior... We must recognize this if we are to understand and choose what we will become as a result of what we have made.”

-Myron Krueger in *Responsive Environments*

HMD (Human Machine Dialectic) Framework

The HMD Framework is a philosophical framework that, by treating machine process as an "other" against which to compare human process, attempts to more clearly understand human beings, while also providing ontology construction for analyzing and optimizing relationships between humans and machines.

HMD differentiation:

HMD point: a process

HMD point elasticity: the range of process flexibility

HMD segment: exists between two HMD points

HMD method-of-increments: move points incrementally and note outcomes

HMD derivative: emerges when an HMD segment is infinitely small

HMD Integration:

The goal is to maximize positive outcomes of human machine relationships. For example, how machine-like should a human process be, and how human-like should a machine process be to achieve optimal interaction?

QUESTIONS:

What can algorithmic processes teach us about ourselves?

What kinds of relationships can humans and machines have?

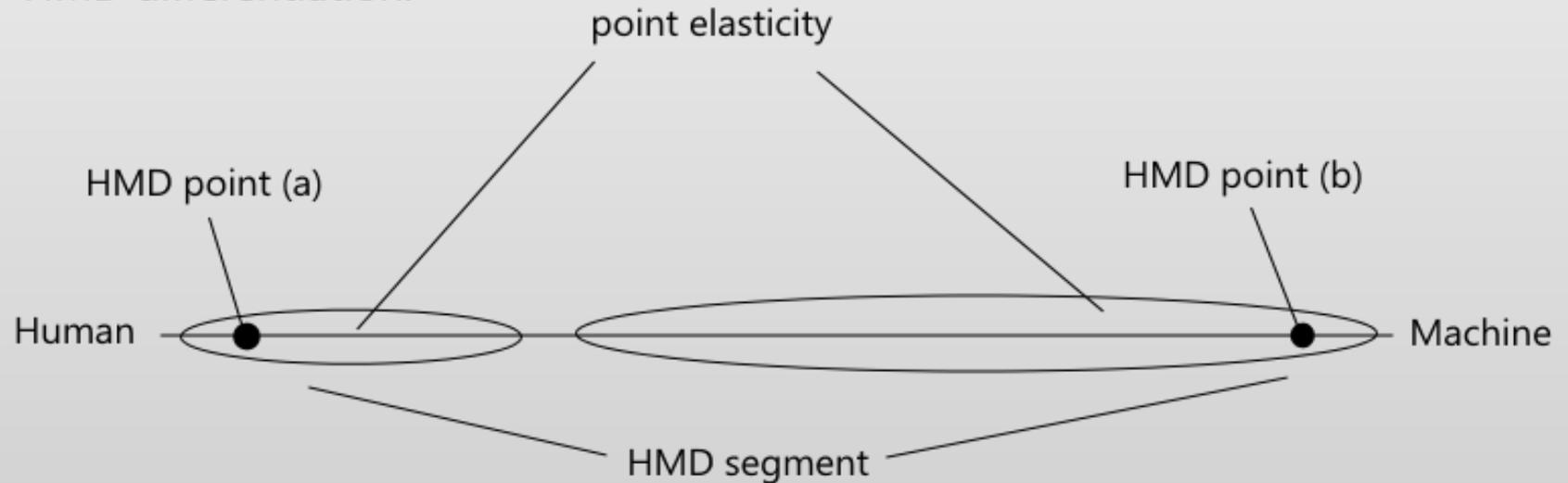
How do humans best leverage machines in creative enterprise?

How are humans different from machines?

How can machines help humans develop the best parts of our nature?

HMD framework

HMD differentiation:



HMD point (a)={human process}

HMD point (b)={machine process}

point elasticity (a)={process flexibility}

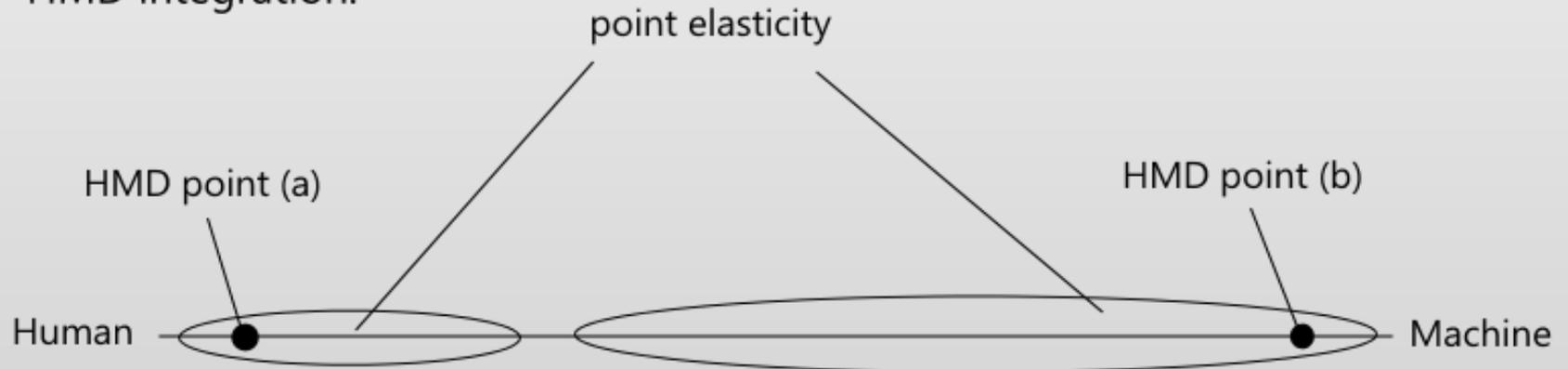
point elasticity (b)={process flexibility}

HMD segment={differences between points}

HMD derivative={emerges when HMD segment becomes infinitely small}

HMD framework

HMD integration:



HMD point (a)={human process}

HMD point (b)={machine process}

point elasticity (a)={process flexibility}

point elasticity (b)={process flexibility}

HMD point (a) + HMD point (b) = {optimal outcome}