Algorithms, SOM, Sonification

M259 DataVis, MAT, 2017 winter

Kohonen Self-Organizing Map

Is an unsupervised neural-network, requires only input data to be trained

Used to automatically find clusters in input data, especially where data elements may be related in a non-linear or associative fashion

Example Input Datafile

Fuzzy	Mammal	Feet?	Useful	Wild	Sounds	Shape	Edible	
0	0	2	1	0	0	2	0	Ladder
1	1	4	1	0	1	3	0	cat
1	1	4	0	1	1	3	0	tiger
0	0	0	0	0	0	0	0	ball
0	0	1	1	0	0	0	0	monitor
0	1	2	1	0	1	3	0	woman
0	1	2	1	0	1	3	0	man
1	1	4	1	0	1	3	0	dog
0	0	0	0	1	0	2	1	goldfish
0	0	3	1	0	0	0	0	tripod
0	0	0	0	0	0	0	1	melon
0	0	0	0	0	0	0	1	onion
0	0	0	0	0	0	0	1	orange
0	0	4	1	0	0	0	0	chair
0	0	3	1	0	0	0	0	stool
0	0	4	1	0	0	1	0	table
0	0	0	1	0	0	0	0	bottle
0	0	0	0	0	0	2	1	banana
0	0	0	0	0	0	0	1	apple

Kohonen Self-Organizing Map

Map Initialization

The map are first initialized to random values.

Map Training

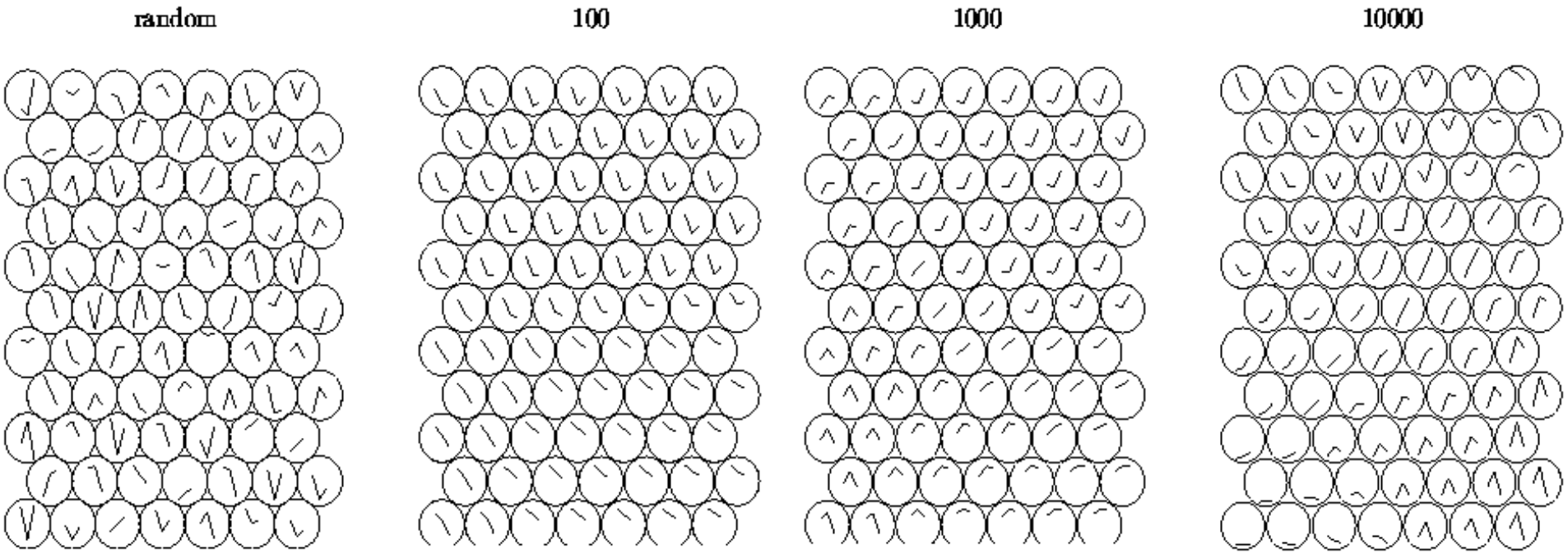
Every node in the map is examined to calculate which one's are most like the input vector.

The radius of the neighborhood of the Best Matching Unit is calculated. Starts large, set to the radius of the lattice and diminishes with each iteration. Radius diminishes with each iteration

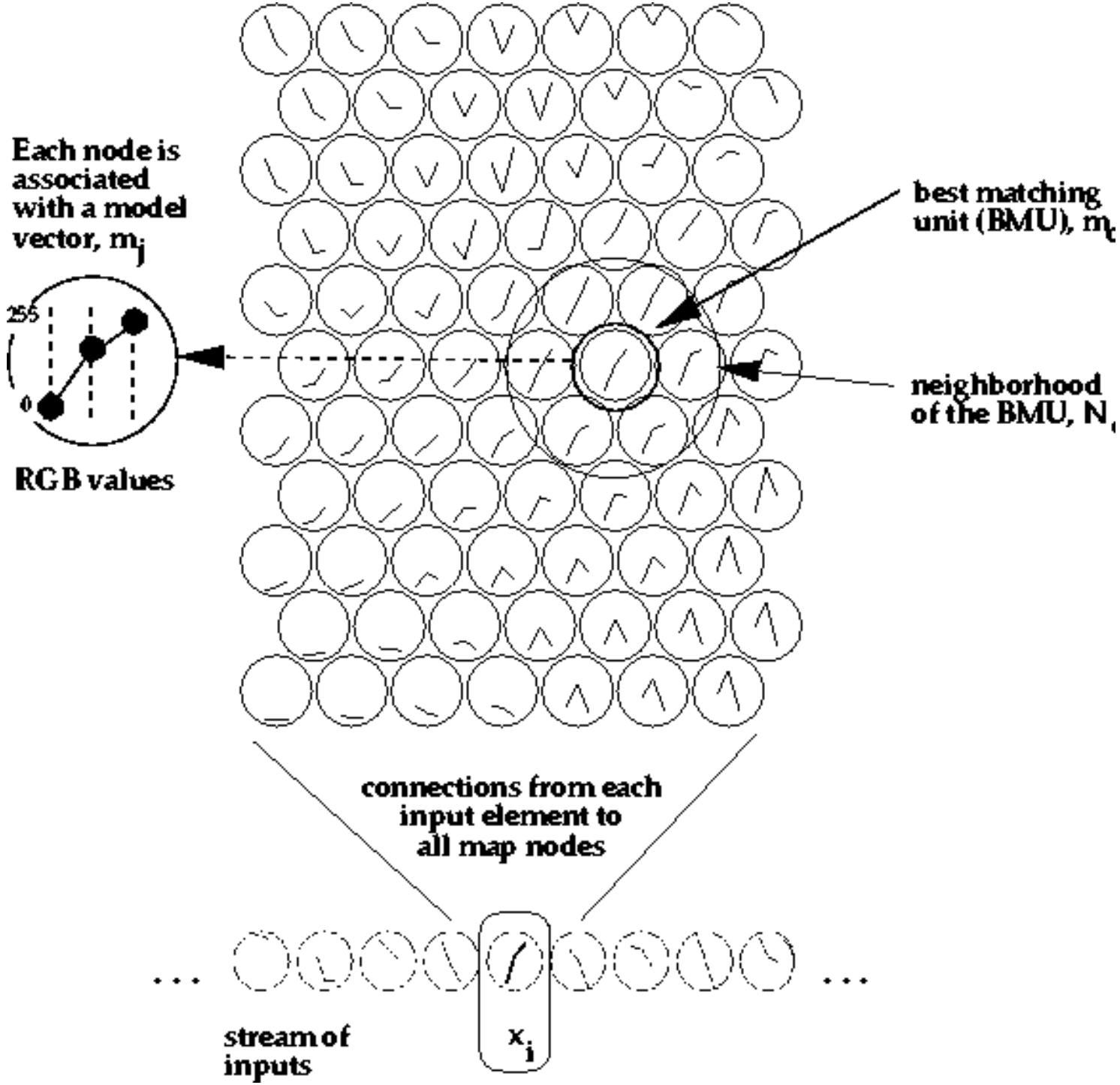
Map Visualization

Calibrate using input samples, attach labels to best matching regions

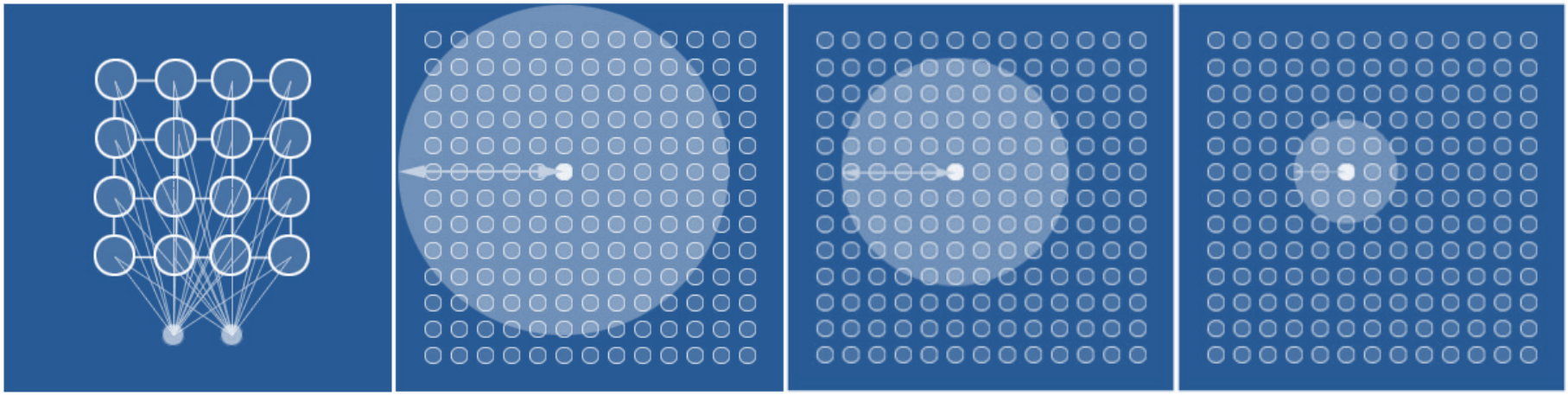
Map Initialization & Training



Timo Honkela Kohonen: https://users.ics.aalto.fi/tho/thesis/


Each node is associated with a model vector, m_j
255
0
RGB values
best matching unit (BMU), m_c
neighborhood of the BMU, N_c
connections from each input element to all map nodes
stream of inputs
x_i

Map Initialization & Training

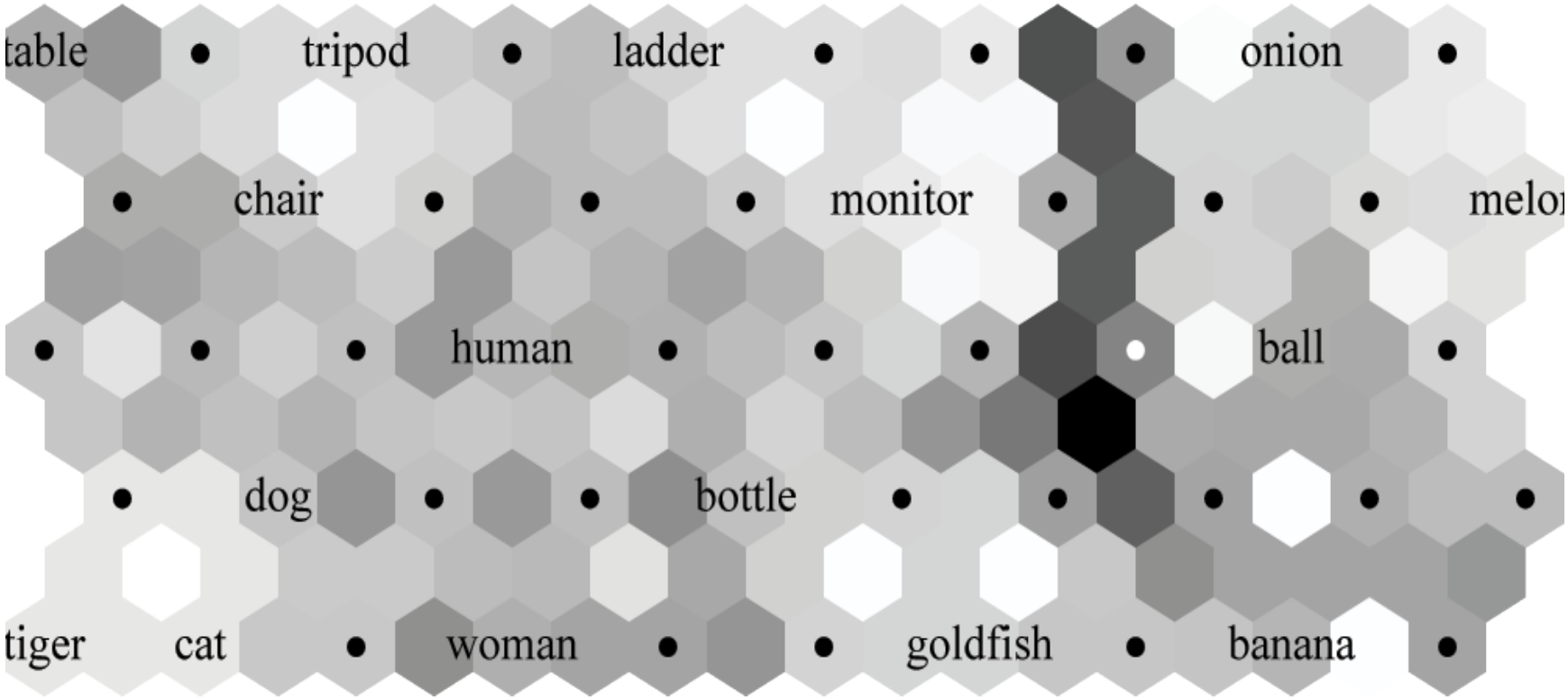


Map Training

Every node in the map is examined to calculate which one's are most like the input vector.

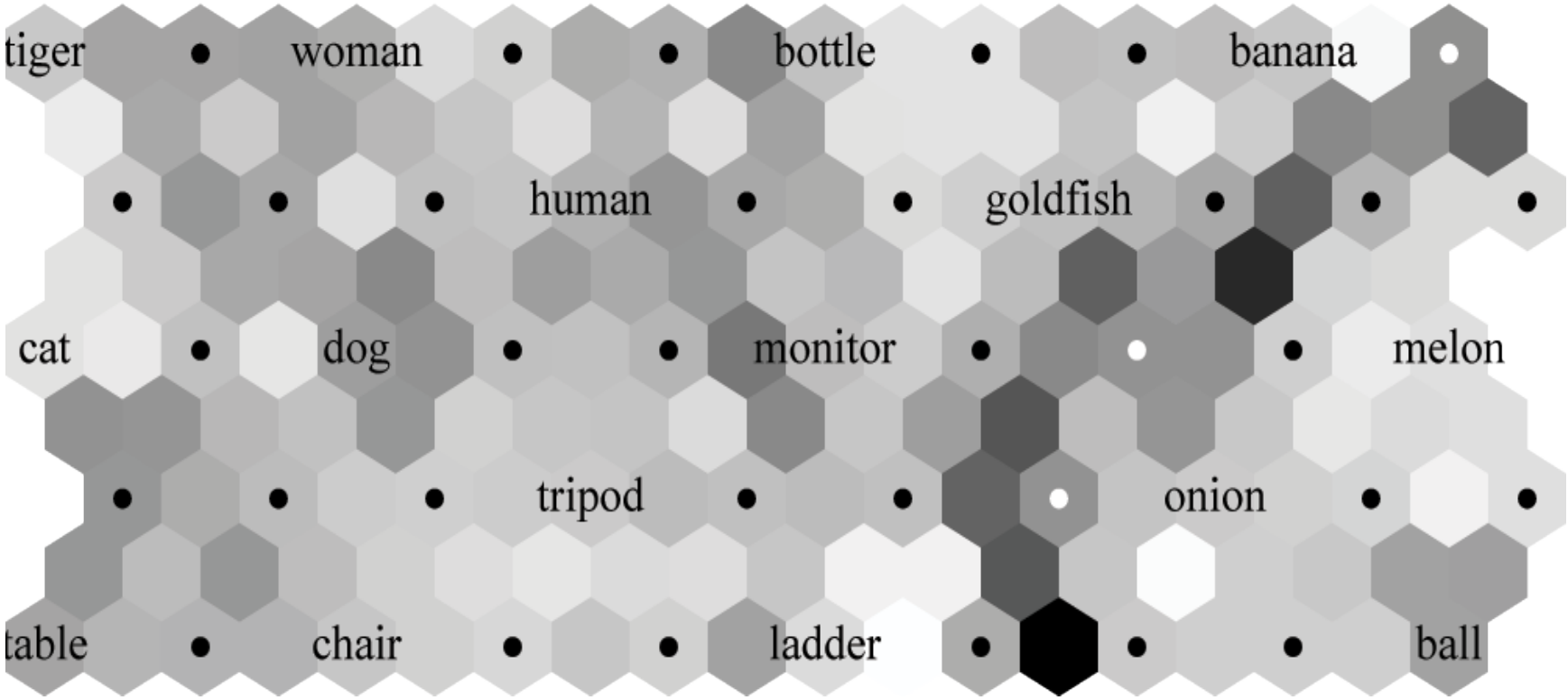
The radius of the neighborhood of the Best Matching Unit is calculated. Starts large, set to the radius of the lattice and diminishes with each iteration. Radius diminishes with each iteration

Map Output − Unified Matrix

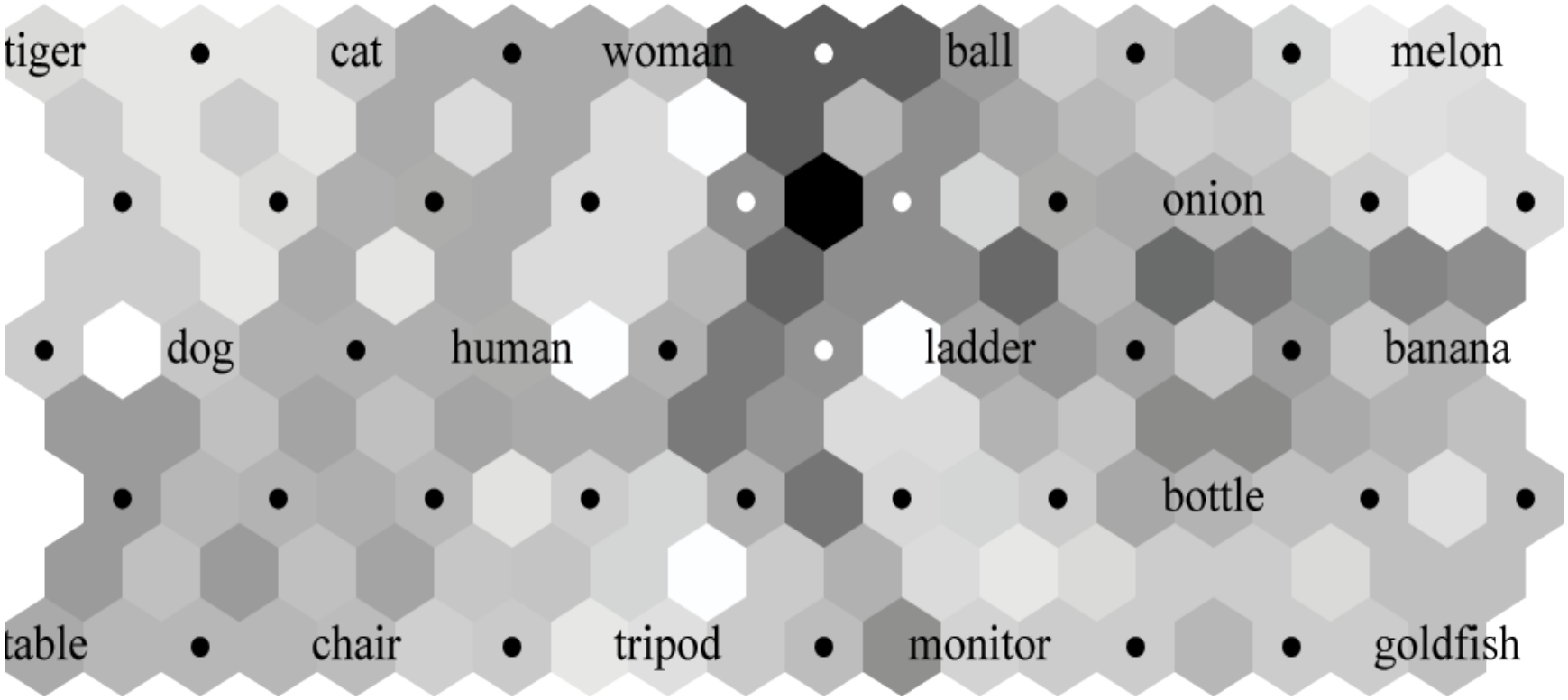


Kohonen Code: http://www.cis.hut.fi/research/som-research/nnrc-programs.shtml

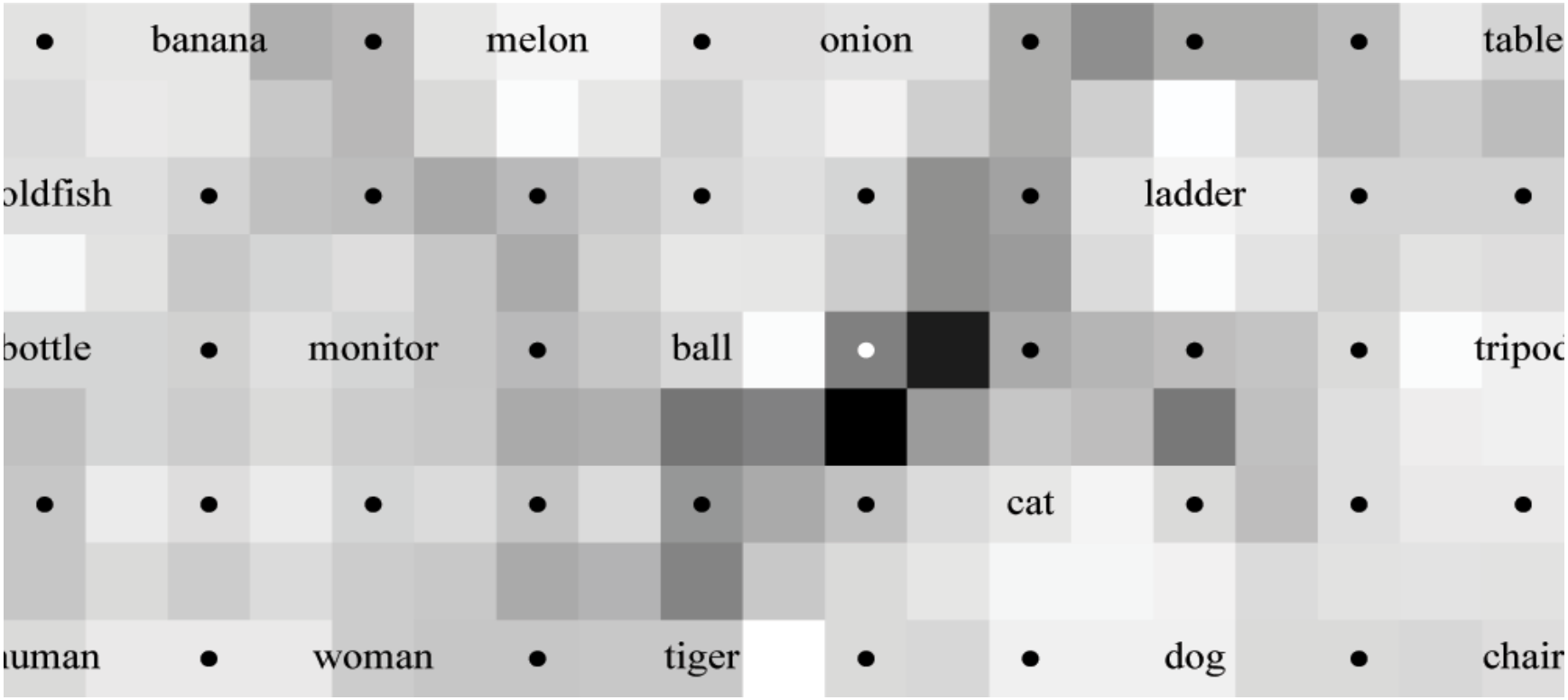
Map Output − Unified Matrix



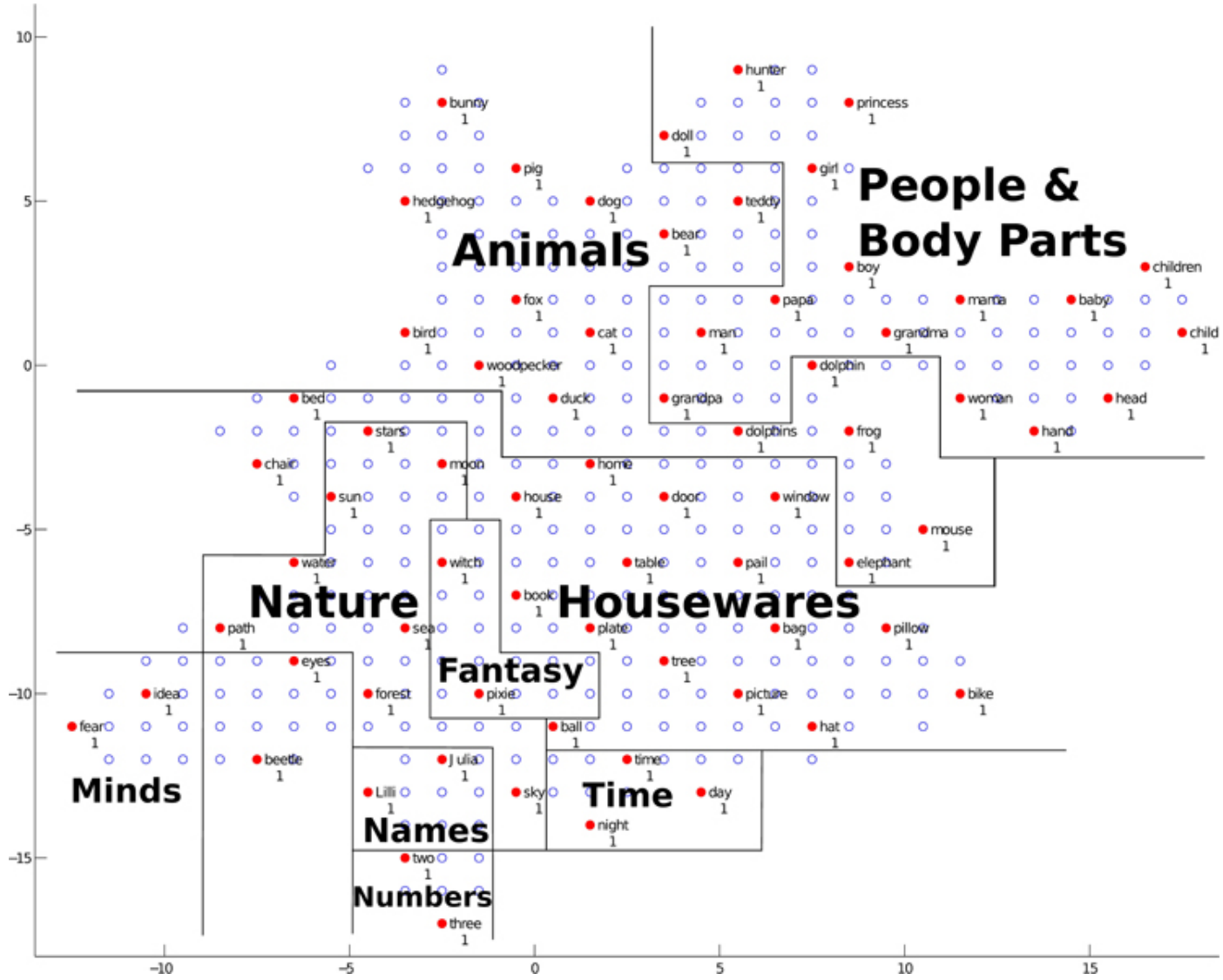
Map Output − Unified Matrix



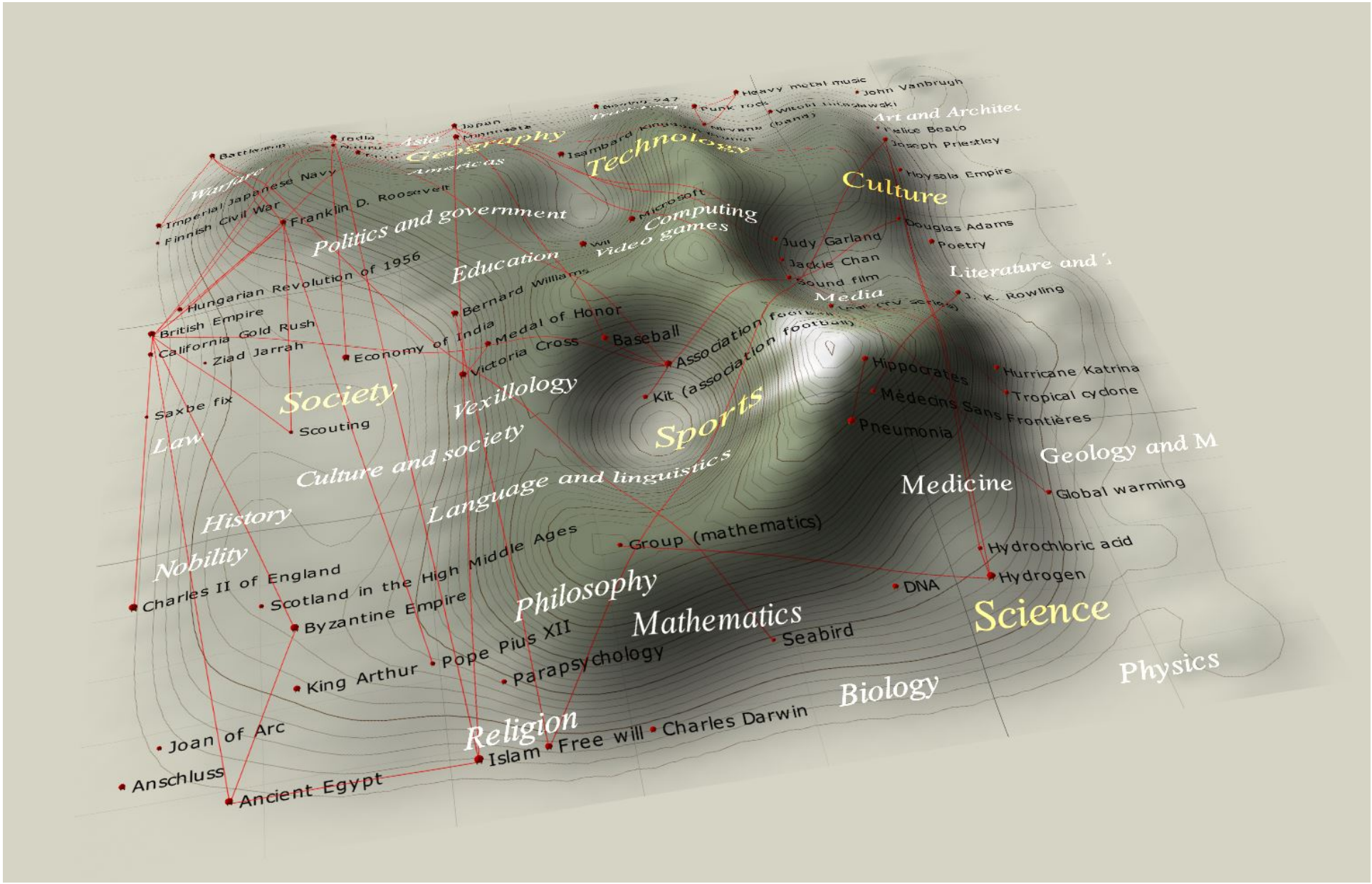
Map Output − Unified Matrix in Grid



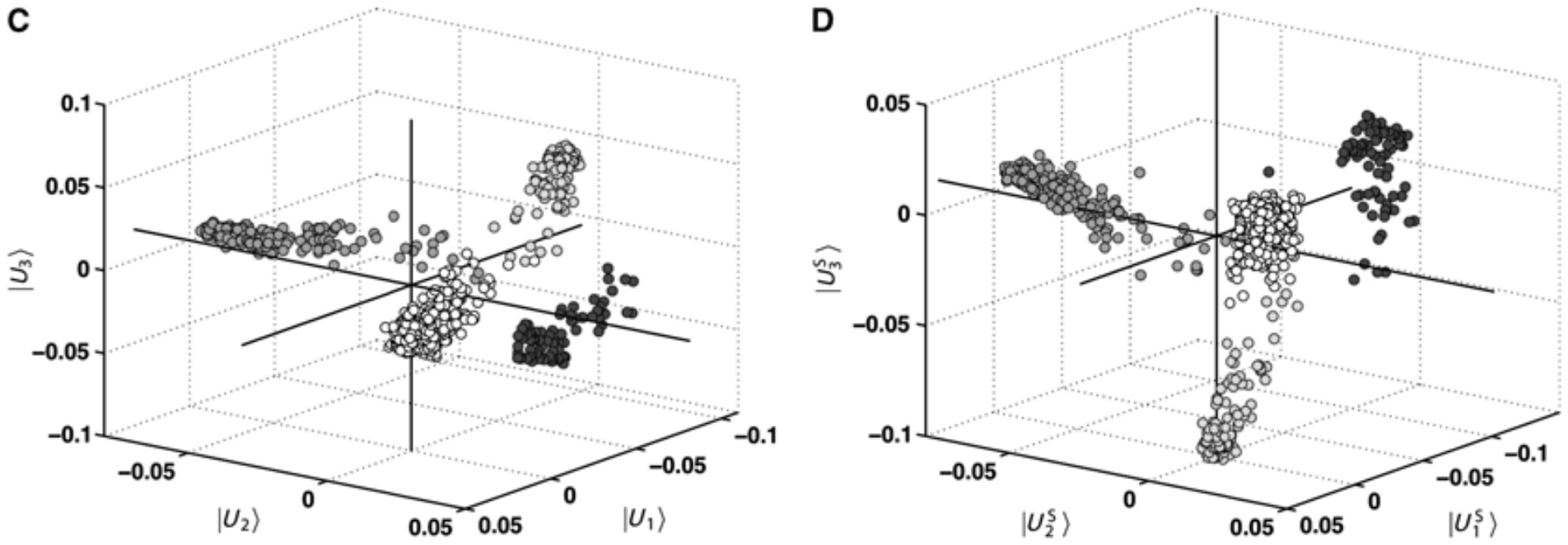
Map Output Grid



Self-Organizing Map: Proximity based on metadata



3D Data Mapping



Audification - Sonification

Audification - Sonification

- **Sonification** − audio to convey information or perceptualize data (geiger counter)
- **Audification** − Representing a sequence of data values as sound. Signal processing is often used to bring out salient data features (wikipedia)
- Users able to detect attributes such as noise, repetitive elements, regular oscillations, discontinuities and signal power in audification of time-series data (Pauletto/Hunt)

http://www.icad.org/Proceedings2005PaulettoHunt2005.pdf

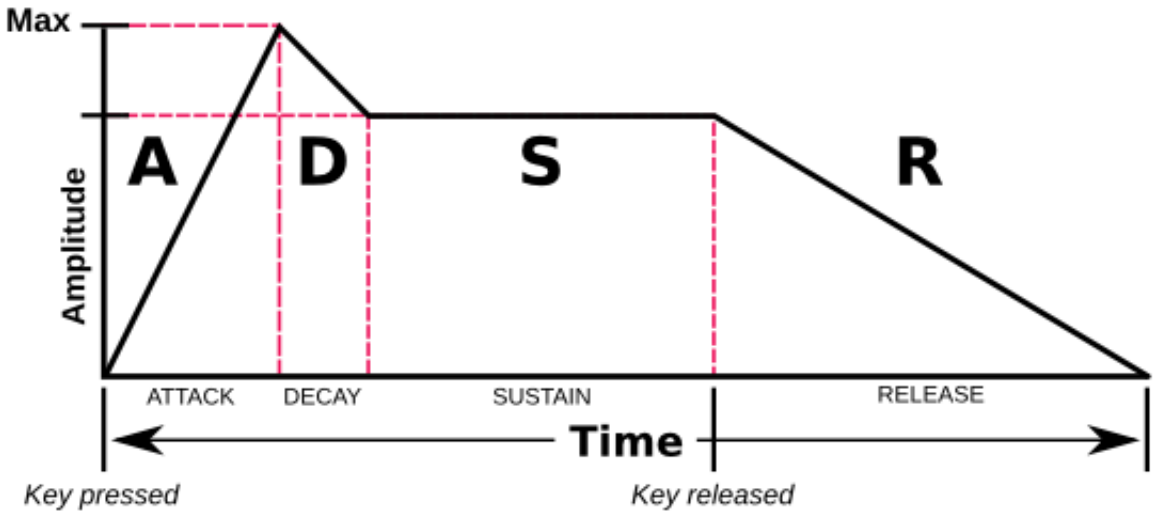
- Karl Yerkes: Sonify checkout counts, mapping the checkout count of each 5-minute interval to the amplitude of each audio sample at 44100 Hz. We heard harmonic distortions characteristic of troubling periods of inactivity. these periods of inactivity turned out to be missing data caused by a bug at the library – which following was resolved.

Data to Sound (Basic Tools)

- **Pitch** − Perceived frequency of a sound
- **Amplitude** − Magnitude of a pulse (as in a sinewave)
- **Tempo** − Beats per minute (BPM) as a way to measure
- **ADSR envelope** − Controls a sounds parameters
- **Frequency Modulation** − Modulating one frequency with another

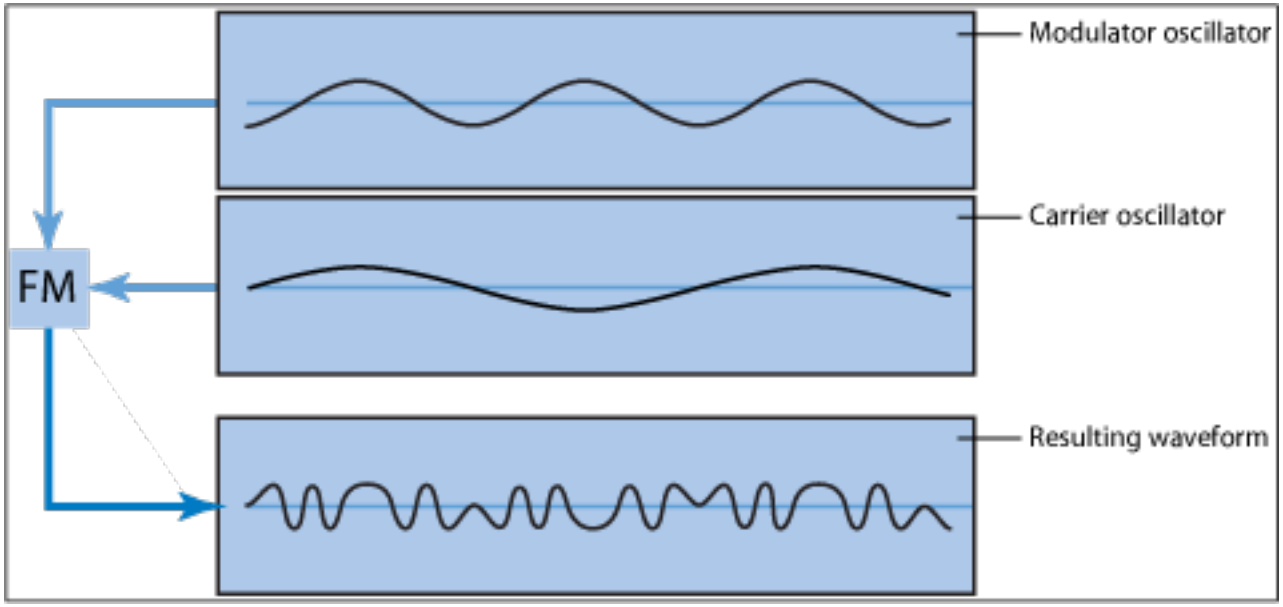
ADSR Envelope

- **Attack Time** − from 0 to peak
- **Decay Time** − from attack level to sustain
- **Sustain Time** − sequence of sounds duration
- **Release Time** − from sustain to 0



Frequency Modulation

- Audio synthesis where the timbre of a simple waveform is changed by modulating its frequency by another resulting in a more complex waveform



Sound in Processing

- **processing.sound** − New sound library with Processing 3. Can play, analyze and synthesize sound
- **Minim** − Audio library for processing

- **Historical Background**: https://processing.org/tutorials/sound/

Examples

Minim

- FM synthesis
- Moog
- noiseTintExample
- frequency example
- vocoderExample
- oscEnvExample

processing.sound

- SoundFile
- Noise
- env // https://processing.org:8443/reference/libraries/sound/Env.html