RegulArt

A visual installation that is reactive to movement, sound volume and its regularity in time

Project general idea

- The installation is based on a particle system of polygons that react to movement and sound:
  - Movement determines the visual characteristics, like creation, direction, speed and position of the polygons
  - Sound is reflected in the behavior of the particles in terms shape regularity
- The particles' lifetime is determined by the presence of sound:
  - Absence causes a progressive death of the particles
  - Presence keeps them alive
- Particles interact between themselves and are attracted by areas in which movement is detected
- Every vertex of each particle is an independent point with respect to the others
  - When regularity is detected in sound the vertices interact among themselves to form a regular shape
  - Otherwise they interact with each other in a chaotic fashion

The project step-by-step

Audio

1. Feature extraction
    - Root Mean Square
    - Zero Crossing Rate
    - Spectral Centroid
    - Spectral Roll-off
    - Spectral Flux
2. Cross-correlation

Video

1. Optical Flow
2. Background removal
3. Clustering

Behavior

1. Particle system
2. Swarm intelligence
    - Cohesion
    - Alignment
    - Separation
3. Mutation

Audio – Context setup and variables

- Channel: Mono
- Sampling Frequency: 44100 Hz
- Buffer size: 2048 samples
- Frame length: 100 ms
- Window history: 10 windows (1 second)

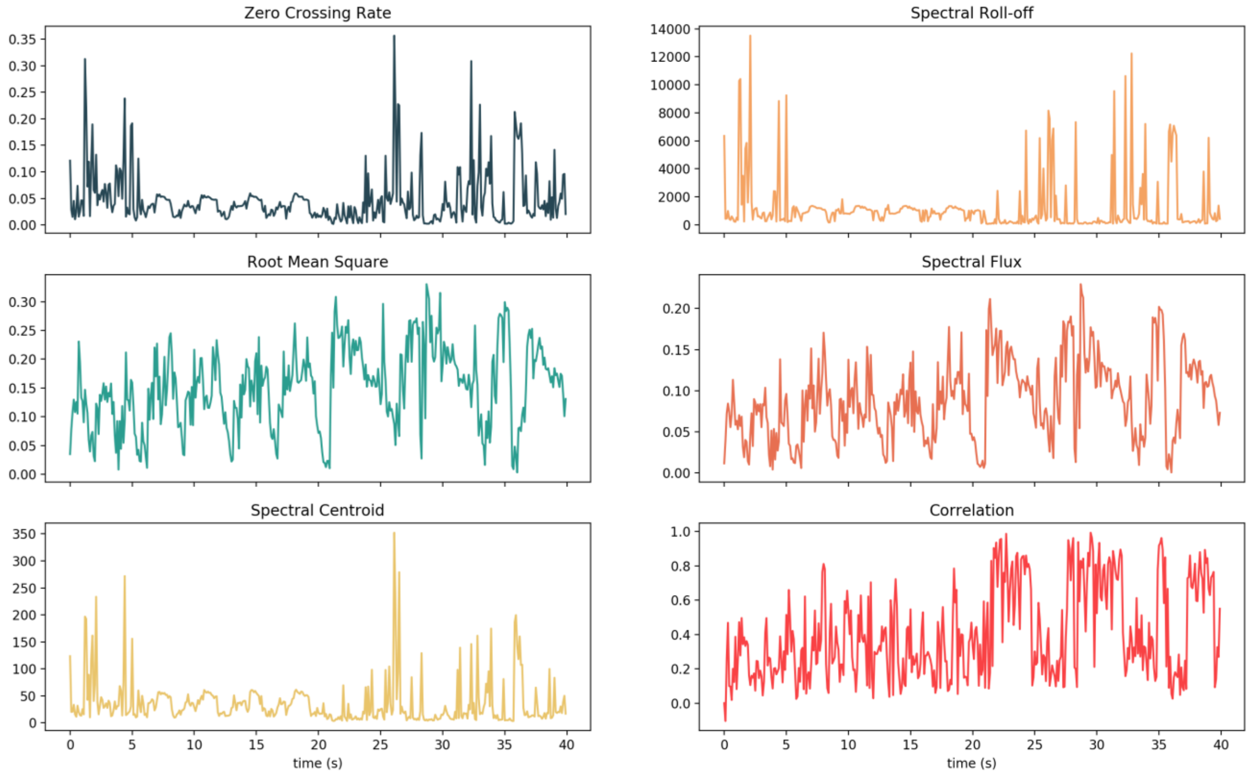
  A set of previous windows is stored in order to be able to have a more reliable comparison with past frames
- Thresholds:

  These values are empirically set
  - Cross-correlation: 0.75
  - Feature vectors euclidean distance: 0.005

Audio – Feature extraction

- Root Mean Square
- Zero Crossing Rate
- Spectral Centroid
- Spectral Roll-off
- Spectral Flux

This set of features is used allows us to evaluate of how different is the current frame from the previous ones and determine an empirical value used to examine regularity



Audio – Cross-correlation

- Audio is gathered in 100 ms windows that are filtered and stored once the feature vector is extracted
- When a new audio window is available, it is normalised and cross-correlated with each element of the windows history vector (if it's populated), resulting in a vector of cross-correlation values, from which the maximum value is taken
- Normalised cross-correlation is used as a regularity metric, since it shows how similar two signals are

Audio – Regularity definition

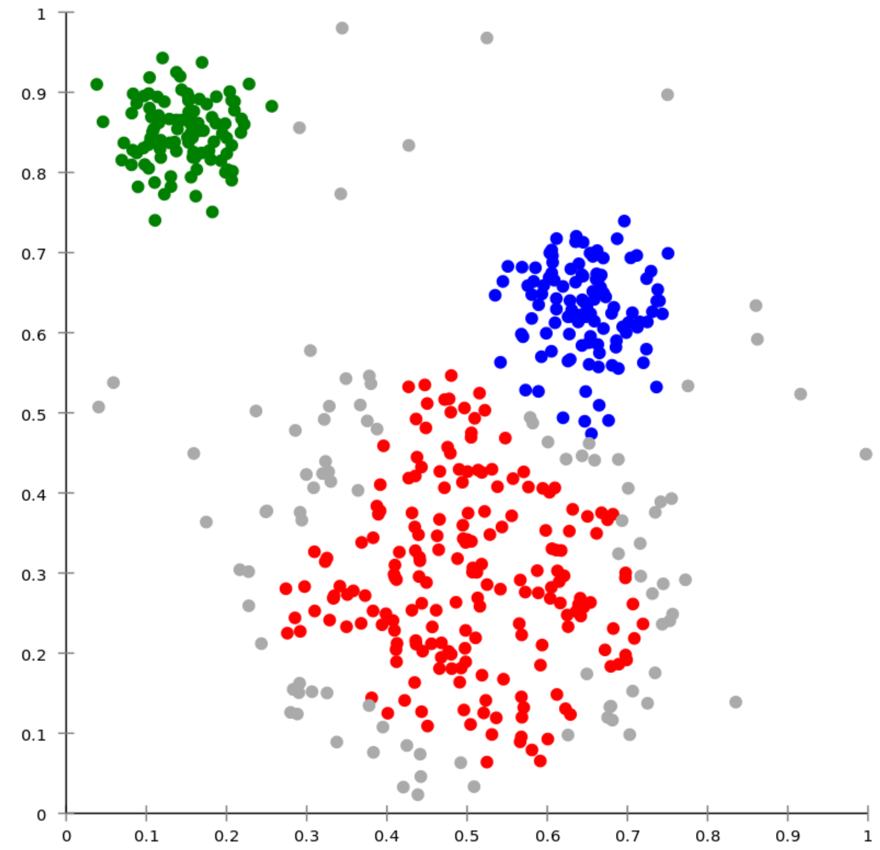
- Regularity is defined on a three conditions basis:
    - If cross-correlation overcomes a threshold the incoming sound is deemed regular; for this reason the correlation threshold should be set quite high.
    - If the above condition is not met, the euclidean distance between the current and the past normalised feature vectors is computed; once again the regularity condition is satisfied by the distance being greater than a given threshold, but a certain degree of cross-correlation is required as well.
    - Otherwise the signal is considered irregular

Video – Optical Flow & Background removal

- Optical Flow is used as a trigger for the generation of particles
- If the average optical flow computed in the area surrounding a pixel overcomes a threshold, said pixel becomes eligible as a creation point for a new particle
  - If background removal has been performed (by clicking on the canvas), motion detection is applied with respect to the background in order to minimise noise; if the threshold is overcome a new particle is created
  - If background removal is not set the creation of a new particles is directly triggered

Video – Clustering

- Clustering is performed with the dbscan clustering function in the Scikit-learn python library
- DBscan distinguishes clusters by identifying areas of high density separated by areas of low density
- This choice is due to the unknown number of clusters beforehand
- The set of points resulting from the optical flow and background removal phase are processed to compute the clusters
- For each cluster a gravity center is computed



Behavior

Particle system

There are two levels of particle systems:

- An individual one, composed by the vertices of a single polygon
- A collective one, which is the entire set of polygons

- Regular:
  - Both level behave in an ordered manner, the individual one forming a regular shape and the collective one arranges itself on a grid
- Non-regular
  - The whole particle system moves according the forces resulting from the collective interaction and moves towards the centers of gravity

Swarm intelligence

- Cohesion

  Keeps the particles together
- Alignment

  Gives a direction to the swarm
- Separation

  Avoids collision between particles

Mutations

When the system behaves in a regular fashion each particle has a certain probability of changing its number of vertices

System architecture

The system operates through three scripts:

- A Processing script: in charge of the visual and behavioral side of the application
- Two Python scripts:
  - Audio script
  - Clustering script

The scripts communicate via the OSC protocol

Thanks for the attention,

De Luca Giorgio

Segato Fabio

Github repo