

There are real sounds, and there are unreal sounds. Sounds that occur in natural real world by material objects are real, and those were all that humans had been able to hear until computers began to make sound solely by computing and directly telling a loudspeaker what sound it should produce at all time.

That we can record real sound and ask a computer to manipulate it means that we can make something real into something unreal. So I questioned what it means musically to hear a piece of sound where the original real sound and processed sound are juxtaposed in a single scene, with smooth transition between the real and unreal regions. I thought that a reasonable implementation of this concept would be one achieved by spectrogram distortion, where in some frequency range the spectrogram is unchanged and in some range frequency-shifted, and the amount of frequency-shift is a linear function of frequency.

Soundscape recording was chosen as the original audio source because it is the sound of real world itself.

In order to represent the psychology that happens inside a listener who is caught up between realness and unrealness of a sound, I imagined a performance where the computer randomly moves around the parameter of the spectral distortion and the player or performer has to always guess what is wrong about the resulting sound and try to revert it back to its unprocessed state by midi-controlling the parameters that computer moves around.

I thought it would be more convincing to set aside a video accompaniment that is also shot from the same place where the soundscape was recorded and equivalently processed, using hue, brightness and time as target variables. If the player could also guess what part of the reality-based source is distorted visually, that would reveal how sense of realness differs among different senses. However for now, due to technical limitations the visual side had to be put aside for later.

For now, I will have to present a simplified version of this idea, where the spectrogram is linearly warped based on two control points on the frequency domain. Computer will randomly move around the two control points, and the player will try to revert that to original as possible with a MIDI controller. On implementing the spectral warp, unwanted noise happened to be unavoidable due to technical limitations. However, the implementation seems like to do the job of continuously warping sound between recognizable and unrecognizable.