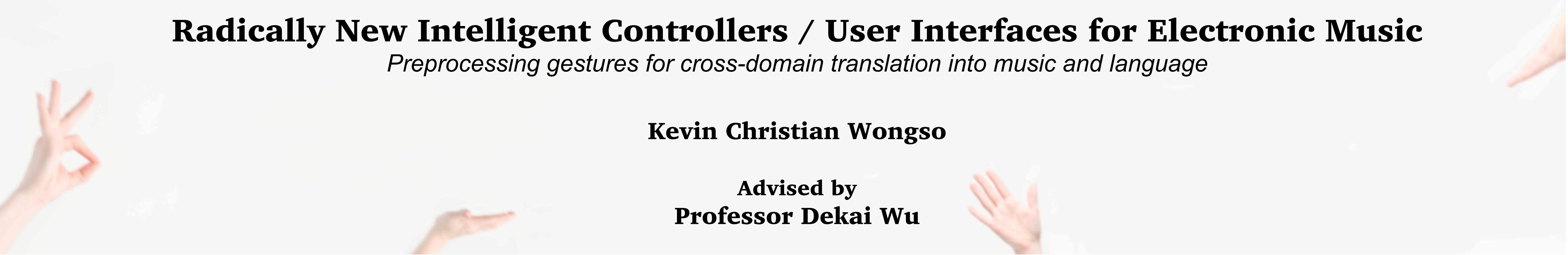


Radically New Intelligent Controllers / User Interfaces for Electronic Music

Preprocessing gestures for cross-domain translation into music and language

Kevin Christian Wongso

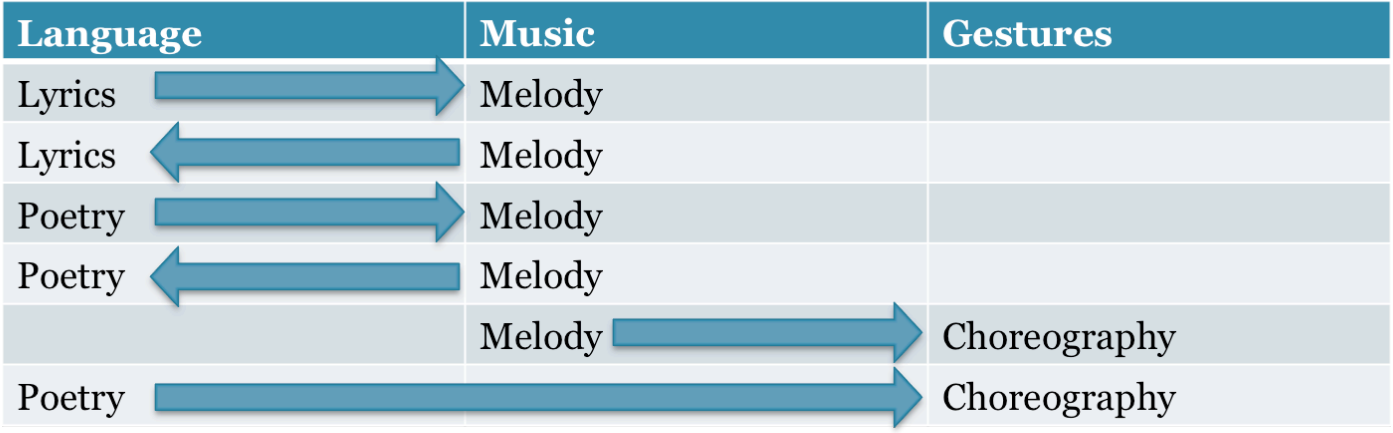
Advised by
Professor Dekai Wu

Our project

This project aims to design and implement a systematic pipeline for the training of interpreters according to a gesture data scheme, and its subsequent use to interpret device-captured low-level gestures, into high-level gestures better suited for cross-domain translation.

Introduction

Gestures, music and language are three domains of expression people translate from and to. All three are also similar in structure, being composed of varying smaller tokens in a particular order. There is high correlation between these domains, which when learnt allows computational translation.



Language	Music	Gestures
Lyrics →	Melody	
Lyrics ←	Melody	
Poetry →	Melody	
Poetry ←	Melody	
	Melody →	Choreography
Poetry →	→	Choreography

However devices usually capture gesture data at a low level, e.g. timestamped coordinates, while we prefer to analyze and translate high-level gestures, e.g. a wave/slap. Additionally there are no standard conventions for high-level gestures; thus systems must design their own gesture data scheme that lists and defines 'valid' gestures the system recognizes and responds to.

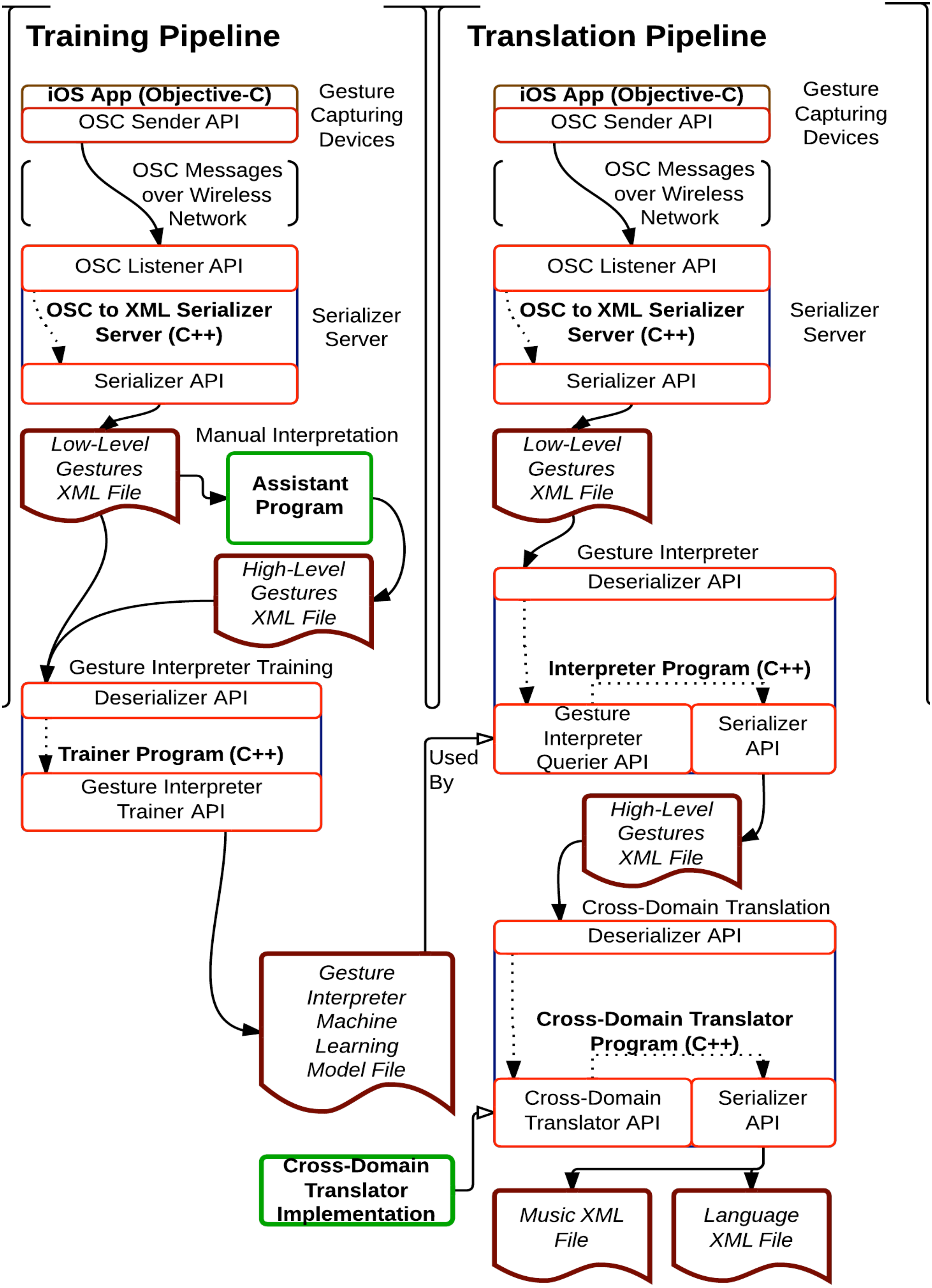
Objectives

The project aims to address the issues and allow gesture data to more easily participate in cross-domain translation by:

- designing and implementing API for a pipeline to train low-level to high-level gesture interpreters, according to a gesture data scheme, and a second pipeline to use previously trained interpreters to interpret low-level gestures to high-level gestures
- designing a cross-domain data representation to which gesture, music, and language data can be encoded without critical information loss (a good criteria of which is the ability to recover the original data from the encoded equivalent); this allows us to represent data from any of the three domains as data from another domain.

Pipeline architecture

Below is our design for the pipelines, with similar functionality grouped into modules with their own API to promote better maintenance, code reusability, and easier future extensibility and modification.


Training Pipeline
iOS App (Objective-C)
OSC Sender API
Gesture Capturing Devices
OSC Messages over Wireless Network
OSC Listener API
OSC to XML Serializer Server (C++)
Serializer API
Serializer Server
Low-Level Gestures XML File
Manual Interpretation
Assistant Program
High-Level Gestures XML File
Gesture Interpreter Training
Deserializer API
Trainer Program (C++)
Gesture Interpreter Trainer API
Gesture Interpreter Machine Learning Model File
Cross-Domain Translator Implementation
Used By
Translation Pipeline
iOS App (Objective-C)
OSC Sender API
Gesture Capturing Devices
OSC Messages over Wireless Network
OSC Listener API
OSC to XML Serializer Server (C++)
Serializer API
Serializer Server
Low-Level Gestures XML File
Gesture Interpreter
Deserializer API
Interpreter Program (C++)
Gesture Interpreter Querier API
Serializer API
High-Level Gestures XML File
Cross-Domain Translation
Deserializer API
Cross-Domain Translator Program (C++)
Cross-Domain Translator API
Serializer API
Music XML File
Language XML File

Low-level gesture XML representation – general description

```
<corpus><domain>Low-Level Gesture</domain>
<header><feature name=”name”>Name of this corpus</feature></header>
<events>
        <event>
                <coordinate axis=”x”>x-coordinate value</coordinate>
                <coordinate axis=”y”>y-coordinate value</coordinate>
                <timestamp>timestamp value</timestamp>
        </event>
        <!-- more events -->
</events></corpus>
```

Cross-domain XML representation – general description

```
<corpus><domain>Language/Gesture/Music</domain>
<header>
        <feature name=”Header Feature Name”>Header Feature Value</feature>
        <!-- more header features  for this corpus-->
</header>
<events>
        <event>
                <!--primary features. gesture: gesture type, music: pitch, language: surface form-->
                <primary label=”name of primary feature”>value of primary feature</primary>
                <optional>
                        <feature name=”optional feature name”>optional feature value</feature>
                        <!-- more optional features for this event-->
                </optional>
        </event>
        <!-- more events -->
</events>
<sentences>
        <sentence id=”sentence id”>
                <begin>beginning timestamp</begin>
                <end>ending timestamp</end>
        </sentence>
        <!-- more sentence definitions -->
</sentences>
</corpus>
```

Results and looking forward

In our project we have tested our implementation with a sample gesture data scheme containing the letters ‘y’ and ‘n’, obtaining accuracies of up to 92%. Future work can test their own gesture data schemes, and use the high-level gestures representation then generated by their pipeline for alignment with representations of other domains, e.g. music, as parallel corpora to train cross-domain translators.