PITCHES 1.0.1

by zplane.development

April 22, 2025

Contents

1 PITCHES Documentation

1.1 Introduction

`PITCHES` is zplane's multi pitch recognition SDK. It analyzes the harmonic content of a music signal and outputs a sequence of midi notes with corresponding time stamps. It is able to transcribe monophonic and polyphonic single instruments as well as polyphonic multi-instrument sources. `PITCHES` assumes that the input signal contains music in Western tonality with 12 pitches per octave (C, C#, D,..., B) and in equal temperament. It detects pitches between MIDI pitch numbers 24 to 96.
`PITCHES` is an offline process which means it requires the complete audio file as input and outputs the note sequence result only after the file has been processed in its entirety. It is *not* an online or real-time multi pitch estimation system. The algorithm is able to process audio files approx. 3.5 times faster than realtime on an Apple M1 Pro processor. The mimimum sample rate is 11025 Hz. There is no restriction on the number of input channels.

This document contains all information you need to get started and use `PITCH↩ ES` to the best of its abilities. At its core are in-depth descriptions of the API through usage examples provided in the PitchesClMain.cpp. This example demonstrate all the functionalities of `PITCHES` and can be used to get started as quickly and easily as possible. What follows is a detailed documentation of the `PITCHES` interface along with all the methods and structs contained within.

1.2 API Documentation

The analysis consists of two stages: a pre-processing stage in which the audio is analyzed, and a processing stage that carries out the actual multi pitch estimation.
The pre-processing stage is based on the push principle: successive blocks of input audio frames are pushed into the Pitches::preProcess() function. It finishes by calling Pitches::finishPreProcess() after the audio file has been entirely pushed into Pitches↩ ::preProcess().

1.2.1 Memory Allocation

The PITCHES SDK does not allocate any buffers handled by the calling application. The input buffer as well as the result objects have to be allocated/created by the calling application.

1.2.2 Naming Conventions

When talking about **frames**, the number of audio samples per channel is meant. For example, 512 stereo frames correspond to 1024 float values (samples). An audio **block** is a sequence of consecutive frames with a given length.

1.2.3 Instance Handling Functions

- **ErrorType Pitches::initialize (float sampleRate, std::size_t fileSizeInFrames, std::size_t numChannels);**
  - Initalizes a `PITCHES` instance.
- **ErrorType Pitches::reset() ;**

- Resets all internal variables and buffers to the default state. The return value indicates whether an error occurred or not.

1.2.4 Process Functions

- **ErrorType preProcess (float const$*$ const$*$ const inputBuffer, int number↩ OfFrames)**
  - Pre-processing function. ppfInputBuffer is an array of pointers to the audio data. inputBuffer[0] is a pointer to the data of the first channel, ppfInput↩ Buffer[1] points to the data of the second channel etc. numberOfFrames specifies the number of frames, i.e. the number of samples in each channel. This function can repeatedly be called with successive chunks of audio until the entire signal has been pushed into `PITCHES`. This function will return an error if it is called after Pitches::finishPreProcess() has been called.
- **ErrorType Pitches::finishPreProcess (bool flushBuffers)**
  - This function has to be called after all audio frames have been pushed into `PITCHES` by Pitches::preProcess(). It can (and needs to) be called only once and will return an error if called before Pitches::preProcess() or after Pitches::process(). Should you plan to add further audio to the buffer via preProcess(...), set the flushBuffers flag to false.
- **ErrorType Pitches::process (std::vector<PitchResultElement>& result)**
  - Performs the actual multi pitch estimation. Returns a vector of pitch sequence elements. This function can only be called after Pitches::process() has been called..

1.2.5 Parameter Retrieving and Setting Functions

- **std::size_t Pitches::getFramesNeeded();**
  - Returns the required number of sample frames in order to process a full block during the next call to Pitches::process()
- **std::size_t Pitches::getMaxFramesNeeded();**
  - Returns the maximum required number of frames needed.

1.3 Command Line Usage Example

The command line example can be executed by the following command

```
PitchesCl -i <inputFile> -r <pitchesResultFile>
```

The complete code can be found in the example source file PitchesClMain.cpp.
In the first step, we create an instance of the `PITCHES` class and initialize:

```
zplane::Pitches pitches;

error = pitches.initialize(inputFile.GetSampleRate(), kBlockSize, inputFile.GetNumOfChannels(
  ));
```

We then read chunks of data from our input file,

```
while (readNextFrame)
{
    // read audio data
    numFramesRead = inputFile.Read (ppfInput, kBlockSize);

    // if there are not enough samples, stop the loop after this iteration
    if (numFramesRead < kBlockSize)
        readNextFrame = false;

    if (verbose)
```

And push each chunk into our preProcess() function.

```
    error = pitches.preProcess (ppfInput, numFramesRead);
```

After the entire file has been read and pushed into PITCHES, we call finishPre↩ Process() once to terminate the preprocessing stage

```
error = pitches.finishPreProcess();
```

We then call PITCHES' process function:

```
error = pitches.process (pitchesResult);
```

We can access the individual pitches of the resulting pitch sequence (e.g. in order to print them on the command line):

```
    cout << "Printing Results!" << endl;

    cout << "Num notes detected: " << pitchesResult.size() << endl;

    for (int i = 0; i < pitchesResult.size(); i++)
    {
        const zplane::Pitches::PitchResultElement element =
  pitchesResult[i];
        cout << "____________________" << endl;
        cout << element.startTimeInS << endl;
        cout << element.endTimeInS << endl;
        cout << element.midiPitchIndex << endl;
        cout << element.velocity << endl;
        cout << "____________________" << endl;
    }
    cout << endl;
```

The above code snippets demonstrated the basic functionality of the PITCHES library.

1.4 Support

Support for the source code is - within the limits of the agreement - available from:

zplane.development

Goerzallee 311
d-14167 berlin
Germany

fon: +49.30.854 09 15.0
fax: +49.30.854 09 15.5

@: info@zplane.de

2 Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

3 Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

4 File Index

4.1 File List

Here is a list of all files with brief descriptions:

5 Namespace Documentation

5.1 zplane Namespace Reference

Classes

- class Pitches

6 Class Documentation

6.1 zplane::Pitches Class Reference

```
#include <Pitches/Pitches.h>
```

Classes

- struct PitchResultElement

Public Types

- enum ErrorType {
  noError, memError, notInitializedError, alreadyInitializedError,
  invalidFunctionParamError, invalidFunctionCallError, canNotBeCalledAtThis↩
  StageError, preProcessNeverCalledError,
  unfinishedPreProcessingError, unknownError, numErrorTypes }

Public Member Functions

- Pitches ()
- ∼Pitches ()
- ErrorType initialize (float sampleRate, std::size_t fileSizeInFrames, std::size_↩
  t numChannels)
- ErrorType preProcess (float const ∗const ∗const inputBuffer, int numberOf↩
  Frames)
- ErrorType finishPreProcess ()
- ErrorType process (std::vector< PitchResultElement > &result)
- ErrorType reset ()
- bool isInitialized ()

Static Public Member Functions

- static const char ∗ getVersion ()
- static const char ∗ getBuildDate ()

6.1.1 Detailed Description

Definition at line 6 of file Pitches.h.

6.1.2 Member Enumeration Documentation

ErrorType `enum zplane::Pitches::ErrorType`

Enumerator

noError	no error occurred
memError	memory allocation failed
notInitializedError	instance has not been initialized yet
alreadyInitializedError	instance has already been initialized
invalidFunctionParamError	one or more function parameters are not valid
invalidFunctionCallError	function call is not allowed

Enumerator

canNotBeCalledAtThisStageError	further audio can't be added via preProcess if finishPreProcessing was called with flushBuffers=true or a data chunk was set, finishPreProcessing can't be called if a data chunk was set
preProcessNeverCalledError	finishPreProcessing() can only be called if preProcess has at least been called once
unfinishedPreProcessingError	finishPreProcessing() has to be called before process() or setKnownDownbeat()
unknownError	unknown error occurred
numErrorTypes	

Definition at line 9 of file Pitches.h.

```
        {
            noError,
            memError,
            notInitializedError,
            alreadyInitializedError,
            invalidFunctionParamError,
            invalidFunctionCallError,
            canNotBeCalledAtThisStageError,
            preProcessNeverCalledError,
            unfinishedPreProcessingError,
            unknownError,
            numErrorTypes
        };
```

6.1.3 Constructor & Destructor Documentation

Pitches() `zplane::Pitches::Pitches ( )`

~Pitches() `zplane::Pitches::~Pitches ( )`

6.1.4 Member Function Documentation

finishPreProcess() `ErrorType zplane::Pitches::finishPreProcess ( )`

Terminates the preprocessing stage.

Needs to be called once before process() can be called.

Returns

Pitches::ErrorType : Returns an error code.

getBuildDate() `static const char* zplane::Pitches::getBuildDate ( ) [static]`

Returns the build date string.

getVersion() `static const char* zplane::Pitches::getVersion ( ) [static]`

Returns major version, minor version, patch and build number of this Pitches version.

initialize()
```
ErrorType zplane::Pitches::initialize (
            float sampleRate,
            std::size_t fileSizeInFrames,
            std::size_t numChannels )
```
Initialize an instance of Pitches. Must be called before using any of Pitches functionality.

Parameters

sampleRate	Sample rate of the input signal in Hertz.
fileSizeInFrames	length of input audio in frames
numChannels	Number of channels in the input signal.

Returns

Pitches::ErrorType : Returns an error code.

isInitialized() `bool zplane::Pitches::isInitialized ( )`

preProcess()
```
ErrorType zplane::Pitches::preProcess (
            float const *const *const inputBuffer,
            int numberOfFrames )
```
Preprocesses a block of audio.

This function can be called multiple times in order to provide successive chunks of the input audio signal.

Parameters

inputBuffer	pointer to the input data chunk. inputBuffer[i] points to the i-th audio channel.
numberOfFrames	The number of audio samples in each audio channel of the provided input data chunk.

Returns

Pitches::ErrorType : Returns an error code.

process() `ErrorType zplane::Pitches::process ( std::vector< PitchResultElement > & result )`

Perform pitches estimation. This is where the major chunk of computational load happens.

Parameters

result	this array contains the pitch result. The order of the elements is determined by the struct element ”startTimeInS”.

Returns

Pitches::ErrorType : Returns an error code.

reset() `ErrorType zplane::Pitches::reset ( )`

Resets the state of the SDK to the same state as directly after calling initialize()

Returns

Pitches::ErrorType : Returns an error code.

The documentation for this class was generated from the following file:

- Pitches/Pitches.h

6.2 zplane::Pitches::PitchResultElement Struct Reference

`#include <Pitches/Pitches.h>`

Public Attributes

- float startTimeInS
- float endTimeInS
- std::size_t midiPitchIndex
- std::size_t velocity

6.2.1 Detailed Description

Struct representing pitch events in a result
Definition at line 34 of file Pitches.h.

6.2.2 Member Data Documentation

endTimeInS `float zplane::Pitches::PitchResultElement::endTimeInS`

Definition at line 37 of file Pitches.h.

midiPitchIndex `std::size_t zplane::Pitches::PitchResultElement::midiPitch↩ Index`

Definition at line 38 of file Pitches.h.

startTimeInS `float zplane::Pitches::PitchResultElement::startTimeInS`

Definition at line 36 of file Pitches.h.

velocity `std::size_t zplane::Pitches::PitchResultElement::velocity`

Definition at line 39 of file Pitches.h.

The documentation for this struct was generated from the following file:

- Pitches/Pitches.h

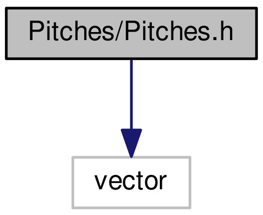
7 File Documentation

7.1 /work/project/docs/docugen.txt File Reference

7.2 Pitches/Pitches.h File Reference

`#include <vector>`

Include dependency graph for Pitches.h:



Classes

- class zplane::Pitches
- struct zplane::Pitches::PitchResultElement

Namespaces

- zplane

Index