ANDROID AUDIORECORD和AUDIOTRACK介

作者 熊熊熊孩子 - 2017-06-11

Android音頻收集和播放(一)

一、文章說明

這是自己第一次通過寫文章的方式來記錄在開發中的一些心得,在這裡也希望這是一個好的開直堅持下去,同時更是希望能幫助到有需要的開發者。

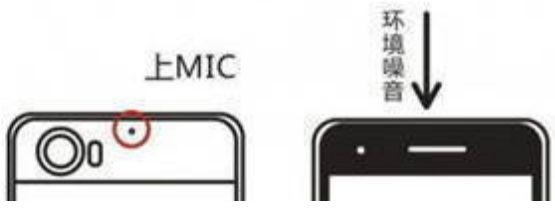
這篇文章主要講述的是Android中使用AudioRecord類和AudioTrack類來進行語音採集和播的知識，在這篇文章中首先介紹的是有關聲音的一些概念性知識，然後介紹聲音的採集，之述Android上回聲消除的相關步驟,最後介紹的是聲音的播放。

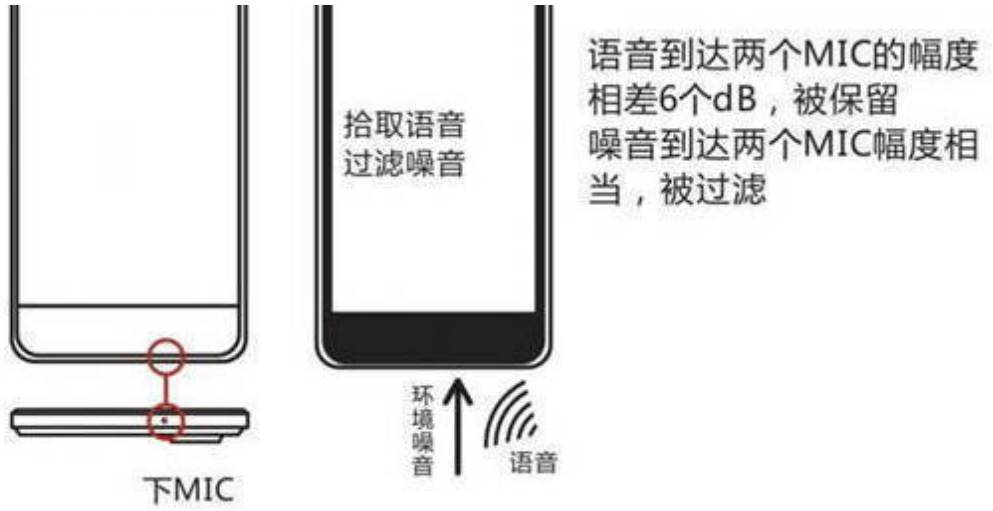
二、概念性知識點

在這裡關於聲音的定義和產生就不再贅述了，如果有對這個感興趣的朋友可以去了解一下，紹幾個聽過但是不是很清楚的概念。

麥克風降噪

大集都知道，在現在這個科技盛行的年代，在打電話時，即使一方的環境有些嘈雜，而在電一方也能聽的清清楚楚，這主要靠的就是手機降噪技術的發展。目前所使用的手機不僅僅有克風，而是有2個甚至是3個(iPhone)，這多出的麥克風就是降噪的關鍵。手機麥克風降噪圖一：





圖一 手機麥克風降噪

下面以兩個麥克風的情況來說一下麥克風降噪，頂部和底部都各有一個。這兩個麥克風都很是兩者的作用是有著非常明顯的區別，其中底部的麥克風是保證清晰通話的，而頂部的麥克來消除噪音的。

由於頂部和底部在通話時和音源(發出聲音的源頭)的距離不同，所以兩個麥克風拾取的音量是有不同的，利用這個差別，我們就可以過濾掉雜訊保留人聲了。在打電話時，兩個麥克風的背景雜訊音量是基本相同的，而記錄的人聲會有6dB左右的音量差。頂端麥收集雜訊後，碼生成補償訊號後就可以用來消除噪音了。

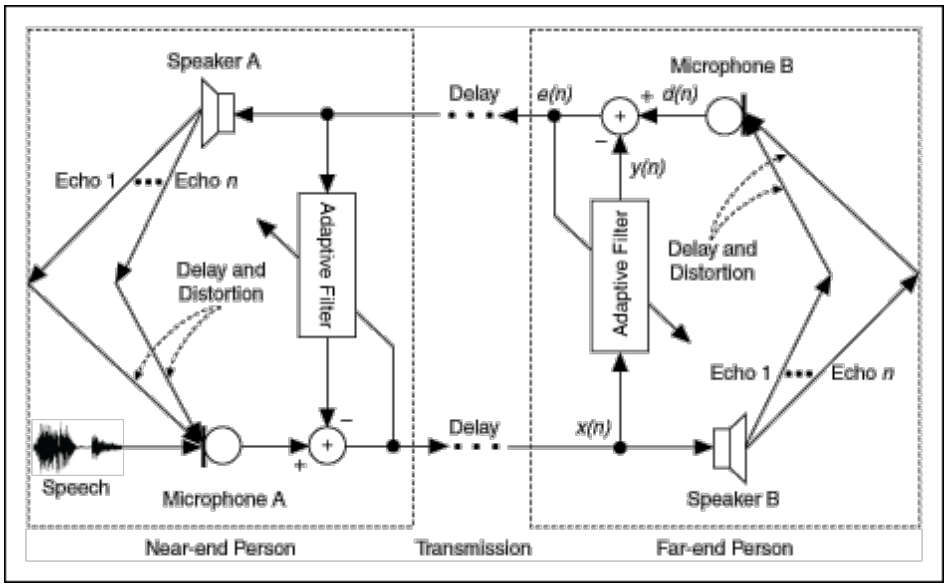
回聲

當聲投射到距離聲源有一段距離的大面積上時，聲能的一部分被吸收，而另一部分聲能要反來，如果聽者聽到由聲源直接發來的聲和由反射回來的聲的時間間隔超過十分之一秒（在15中，距聲源至少17米處反射），它就能分辨出兩個聲音這種反射回來的聲叫“回聲”。如果聲知，當測得聲音從發出到反射回來的時間間隔，就能計算出反射面到聲源之間的距離。例如（20℃）時空氣中的聲速是343米每秒，所以站在聲源處的人要聽到回聲需要障礙物到聲源

回聲消除

很多時候在收集語音的同時也會播放著聲音，這就要在語音收集的時候就需要對採集的聲音
聲消除。當在語音收集的時候，如果還同時播放著聲音的話，就不能保證採集的聲音不包括
放的聲音，在這種情況下採集的語音即包括原聲又包括回聲，在這樣的惡性迴圈下，就會使
越來越多，最後出現嗡鳴聲。

回聲消除就是在麥克風錄製外音的時候去除掉手機自身播放出來的聲音，這樣就將播放的聲
集的聲音中過濾出去，從而就避免了回聲的產生。圖二很好展示了回聲消除的機制。



圖二 回聲消除

在近端，麥克風會採集到揚聲器播放出來的遠端聲音，假設這路聲音為y(n)，當然由於需要
傳來播放出來，我們當然能得到遠端傳來的聲音訊號，假設這路聲音為x(n)。不難發現x(n)經
器的播放，然後經過空氣的傳播，最後被麥克風採集，然後變為y(n)，x(n)和y(n)具有明顯的
性。假設麥克風採集到的總聲音訊號為z(n)，這時候需要通過自適應濾波器根據x(n)找出z(n
y(n)，然後從z(n)中過濾掉y(n)。

SHARES

麥克風的原理是將將採集的聲音轉換為類比電訊號，之後將類比電訊號數字化，也就是用高
表示的訊號(計算機能識別的訊號),在Android中有一個AudioRecord類就能錄製語音，並將
換為PCM數據，聲音在經過麥克風轉換為類比電訊號並最終又轉換為PCM數據，在轉換為P
時就要依賴於三個參數，分別是：聲道數、採樣位數和採樣頻率。

聲道數

很好理解，有單聲道和立體聲之分，單聲道的聲音只能使用一個喇叭發聲（有的也處理成兩
輸出同一個聲道的聲音），立體聲的PCM可以使兩個喇叭都發聲（一般左右聲道有分工）
受到空間效果。

採樣位數

即採樣值或取樣值（就是將採樣樣本幅度量化）。它是用來衡量聲音波動變化的一個參數，
說是音效卡的解析度。它的數值越大，解析度也就越高，所發出聲音的能力越強。

在計算機中採樣位數一般有8位和16位之分，但有一點請大家注意，8位不是說把縱坐標分成
而是分成2的8次方即256份； 同理16位是把縱坐標分成2的16次方65536份。

採樣頻率

即取樣頻率，指每秒鐘取得聲音樣本的次數。採樣頻率越高，聲音的質量也就越好，聲音的
就越真實，但同時它占的資源比較多。由於人耳的解析度很有限，太高的頻率並不能分辨出
16位音效卡中有22KHz、44KHz等幾級，其中，22KHz相當於普通FM廣播的音質，44KHz已
CD音質了，目前的常用採樣頻率都不超過48KHz。

既然知道了以上三個概念，就可以由下邊的公式得出PCM檔案所佔容量：

存儲量=(採樣頻率×採樣位數×聲道×時間)/8(單位：位元組數)

四、**Android**聲音錄製

Android中使用AudioRecord和MediaRecorder進行音頻的錄製，這裡介紹AudioRecord類用，有朋友對MediaRecorder感興趣的就去學習一下。根據上邊的介紹可知，需要給Audio傳入採樣頻率、採樣位數和聲道數，除此之外還需要傳入兩個參數，一個是聲音源，一個是大小。在這裡緩衝區大小不能小於最小緩衝區大小(下面有介紹)。

許可權

在Android中進行音頻的錄製是需要相應的許可權的。在Android 6.0及以上要動態的申請許

音頻源

下面是Android所支援的音頻源：

```
/**預設聲音**/

public static final int DEFAULT = 0;

/**麥克風聲音*/

public static final int MIC = 1;

/**通話上行聲音*/

public static final int VOICE_UPLINK = 2;

/**通話下行聲音*/
```

/**通話上下行聲音*/

public static final int VOICE_CALL = 4;

/**根據攝像頭轉向選擇麥克風*/

public static final int CAMCORDER = 5;

/**對麥克風聲音進行聲音識別，然後進行錄製*/

public static final int VOICE_RECOGNITION = 6;

/**對麥克風中類似ip通話的交流聲音進行識別，預設會開啟回聲消除和自動增益*/

public static final int VOICE_COMMUNICATION = 7;

/**錄製系統內置聲音*/

public static final int REMOTE_SUBMIX = 8;

在我的項目中音頻源來自於麥克風聲音。

緩衝區大小

設置完音頻源接下來就要設置緩衝區大小。麥克風採集到的音頻數據先放置在一個緩衝區裡後我們再從這個緩衝區裡面讀取數據，從而獲取到麥克風錄製的音頻數據。在Android中不道數、採樣位數和採樣頻率會有不同的最小緩衝區大小，當AudioRecord傳入的緩衝區大小小緩衝區大小的時候則會初始化失敗。大的緩衝區大小可以開啟更為平滑的錄製效果，相應帶來更大一點的延時。如下圖所示展示了緩衝區在音頻錄製和獲取中所處的位置：

圖三 緩衝區大小

這裡提到的最小緩衝區的大小可以由下邊的代碼來獲得：

AudioRecord.getMinBufferSize(frequency, channelConfiguration,audioEncoding);

當獲取最小緩衝區大小失敗的時候，會返回相應的負數錯誤碼。

AudioRecord的初始化

下面的代碼是AudioRecord的初始化處理

public AudioRecord(int audioSource, int sampleRateInHz, intchannelConfig, int audioF
int bufferSizeInBytes) throwsIllegalArgumentException

這個構造方法中的參數上邊已經有了講解。當初始化失敗時也就是傳入的參數不匹配時就會
常。如果想知道是否初始化成功的話可以獲取一下AudioRecord的一個狀態量，通過getSta
可以獲取，當返回為STATE_UNINITIALIZED表示未成功初始化，當返回為STATE_INITIALIZ
已經成功初始化了，初始化成功後就可以進行讀取緩衝區中的音頻數據的操作了。

讀取數據

AudioRecord使用read()方法進行數據的讀取操作，正如下面的這個方法：

public int read(@NonNull byte[] audioData, int offsetInBytes,int sizeInBytes) {

}

當從緩衝區獲取音頻數據失敗時，會返回負數的錯誤碼。

參數選擇

眾所周知Android的廠商非常多，這就給開發者們製造了很多的麻煩，最為迫切解決的也就
性的問題，在這方面Android推薦在錄音時使用的參數為：

sampleRateInHz = 44100;//採樣率

channelConfig = AudioFormat.CHANNEL_CONFIGURATION_MONO;//音道數

audioFormat = AudioFormat.ENCODING_PCM_16BIT//採樣位數

五、**Android**回聲消除

在Android中回聲消除可以通過三種方式進行處理：1、通過VOICE_COMMUNICATION模式
音，自動實現回聲消除；2、利用Android自身帶的AcousticEchoCanceler進行回聲消除處理
使用第三方庫（Speex、Webrtc）進行回聲消除處理(在項目中我所使用的就是Speex進行回
的)。

通過第一種方式進行回聲消除，需要將AudioManager設置模式為MODE_IN_COMMUNICA
還需要將麥克風開啟。有一點需要特別注意，音頻採樣率必須設置**8000**或者**16000**，通道數
為**1**個**(**別問為什麼，這就是規定**)**。

AudioManager audioManager
=(AudioManager)mContext.getSystemService(Context.AUDIO_SERVICE);

audioManager.setSpeakerphoneOn(true);

使用AcousticEchoCanceler過程比較簡單，錄製聲音的時候可以通過AudioRecord得到AudioSessionId，在建立AudioTrack的時候也可以傳入一個AudioSessionId，這時候將這個AudioSessionId傳入AcousticEchoCanceler，那麼AcousticEchoCanceler將根據之前講過的消除的原理進行回聲消除。代碼如下：

```
 private void initAec(int audioSessionId) {
    if(!AudioAecUtils.isAcousticEchoCancelerApproved()) {
        aecSwitch.setEnabled(false);
        return;
    }
    aec = AcousticEchoCanceler.create(audioSessionId);
    if (aec == null) {
        Log.e(TAG, "AcousticEchoCanceler.create failed");
        aecSwitch.setEnabled(false);
        return;
    }
}

private boolean setEnableAec(boolean enable) {
    if (aec == null) {
        return false;
    }
    int ret = aec.setEnabled(enable);
    if (ret != AudioEffect.SUCCESS) {
        Log.e(TAG, "AcousticEchoCanceler.setEnabled failed");
        return false;
    }
    if(enable) {
        Log.d(TAG, "Aec On");
    } else {
        Log.d(TAG, "Aec Off");
    }
    return true;
}
```

AcousticEchoCanceler回聲消除代碼

當使用Speex或者WebRtc等三方庫進行回聲消除的時候，需要將採集到的音頻數據傳入作為

據，需要將此刻播放的音頻數據傳入作為參考數據，然後還需要傳入一個延時間隔，這樣第
就能工作，從而得到回聲消除後的聲音。因為播放的聲音需要傳播，而且麥克風採集聲音還
的緩衝區，因此需要傳入一個延時間隔。關於Speex和Webrtc在github上能找到相應的And
ndk庫。有興趣的朋友可以實際動手進行測試一下。

六、**Android**聲音播放

前面介紹了Android的音頻錄製，下面介紹一下音頻的播放，同樣Android有兩個音頻播放的
MediaPlayer和AudioTrack，這裡介紹AudioTrack類，其實MediaPlayer在framwork層也
了AudioTrack來進行解碼生成PCM，然後代理給AudioTrack。

StreamType(指定在流的類型**)**

這個在構造AudioTrack的第一個參數中使用。這個參數和Android中的AudioManager有關
及到手機上的音頻管理策略。Android將系統的聲音分為以下幾類常見的：

STREAM_ALARM：警告聲

STREAM_MUSCI：音樂聲，例如music等

STREAM_RING：鈴聲

STREAM_SYSTEM：系統聲音

STREAM_VOCIE_CALL：電話聲音

模式類型

這個在構造AudioTrack的最後一個參數中使用。AudioTrack中有MODE_STATIC和MODE_S

的缺點就是JAVA層和Native層不斷地交換數據，效率損失較大。而STATIC方式表示是一開始時候，就把音頻數據放到一個固定的buffer，然後直接傳給audiotrack，後續就不用一次次了。AudioTrack會自己播放這個buffer中的數據。這種方法對於鈴聲等體積較小的檔案比較

AudioTrack的初始化

下面的代碼是AudioTrack的初始化處理

public AudioTrack(int streamType,int sampleRateInHz, int channelConfig,

int audioFormat,int bufferSizeInBytes,int mode)throws IllegalArgumentException

{}這個構造方法中的參數上邊已經有了講解。當初始化失敗時也就是傳入的參數不匹配時就異常。

寫入數據

AudioTrack使用write()方法將數據寫入到audiotrack中，正如下面的這個方法：

public int write(byte[] audioData,int offsetInBytes, int

sizeInBytes) {}offsetInBytes是指要播放的數據是從參數audioData的哪個地方開始。

七、總結

好了，這個第一篇就寫到這裡了，希望朋友們能對Android錄音和播放有一定的了解。後續新。

八、相關連接

www.jianshu.com/p/2cb75a71009f

www.jianshu.com/p/bee958826a9e

熊熊熊孩子