Parte I
Sintesis basada en muestras

1. Introduccion

La sintesis basada en muestras consiste en la realizacion de un audio a partir de un numero reducido de archivos de audios que ya se tenian almacenadas.Los sonidos que puedan llegar a requerirse que no se tengan almacenados(ya sea porque no coincide la duracion,pitch,volumen,etc) se generan a partir de la modificacion de las muestras que se tienen almacenadas.La ventaja de este tipo de sintesis es que permite sintetizar canciones que suenan muy similares a lo que serian si se hubieran realizado con instrumentos fisicos.

A lo largo de esta seccion se presenta cuales son las principales dificultades de la sintesis basada en muestras y los algoritmos que existen para lidiar con estas dificultades.Asimismo, se compara las diferencias, ventajas y desventajas de cada uno de estos algoritmos, y a partir de esto se discute cual fue el razonamiento que llevo al sintetizador implementado.

2. Time Scaling Algorithms

El mayor problema a enfrentar en este tipo de sintesis es como generar los sonidos que se necesitan para los cuales no se tiene una muestra de audio.Por ejemplo, si se tiene un audio almacenado de como suena la tecla correspondiente al do_4en un piano con una duracion de un segundo.Es posible obtener la nota correspondiente a un semitono mas alto $do_4\#$ de la misma duracion que la muestra almacenada?

Para ajustar la 'altura'(pitch),que tan grave o aguda se escucha la nota, se puede remuestrear el audio de una nota que se tenga a una frecuencia de muestreo distinta y reinterpretar la senal obtenida con la misa frecuencia de muestreo original.Esto se puede describir matematica como el siguiente sistema:

$$y(nT) = x(n.m.T) \tag{1}$$

Se pueden distinguir dos casos posibles, cuand m<0 se toman muestras...

El problema con remuestrear el audio es que ademas de cambiar el contenido armonico de la senal tambien cambia su extenson en el tiempo.Para lograr de independizar el escalamiento en frecuencia del escalamiento en tiempo de una senal existen lossdenominados algoritmos de TSM(TimeScaleModification).Si se puede lograr cambiar la extension temporal de una senal de audio sin distorsionar su contenido armonico(conservar su pitch) entonces simplemente se puede remuestrear una muestra almacenada para obtener el pitch deseado y luego aplicar un algoritmo TSM para corregir la extension temporal a la cual se desea.

Hay un par de nociones basicas que se suelen utilizar en todos los algoritmos TSM. En general estos algoritmos consisten en dividir la senal de entrada en bloques centrados en una determinada posicion(comunmente se los denomina 'bloques de analisis), aplicar algun tipo de operacion matematica sobre estos bloques y copiarlos a otro bloque en la salida(denominnados 'bloques de sintesis'). Se define la funcion de mappeo(concida generalmente como 'time stretch function') a:

$$\tau(nT) = f(nT) \tag{2}$$

$n\epsilon\,[0, N-1]$, donde N es el numero de muestras de la senal de entrada

En principio la funcion de mappeo τ no tiene porque ser una funcion lineal,solo se asume que la funcion es monotona creciente y que $\tau(0) = 0$.

La funcion de mappeo definida en (2) sirve como parametro para todos los algoritmos de TSM que se desarrollaran a continuacion.El rol de esta funcion es la de asignar que posicion le corresponde a la salida a un valor de la entrada tomado en el timpo nT.

Existen una gran variedad de algoritmos TSM,generalmente se distinguen entre los que trabajan en el dominio de la freccuencia y los que trabajan en el dominio del tiempo. A continuacion se presntaran y discutiran los tres

algoritmos implementados en nuestro sintetizador. Los primeros dos (OLA y WSOLA) corresponden a algoritmos que trabajan en el dominio del tiempo, el ultimo y tercer algoritmo(Phase Vocoder) trabaja en el dominio de la frecuencia.

2.1. Overlap and Add(OLA)

OLA corresponde al algoritmo TSM mas simple y posiblemente el mas intuitivo.Existen una gran variedad de algoritmos TSM que se basan en una version modificada del metodo OLA[1].

En definitiva consiste en dividir la senal que se desea modificar en segmentos(denominados grains o slots) de una longitud determinada mediante la aplicacion de ventanas,y luego copiar estos segmentos a una ubicacion determinada en el output dada por una funcion de mappeo(la misma que se definio en (2)). Graficamente el esquema seria el siguiente:

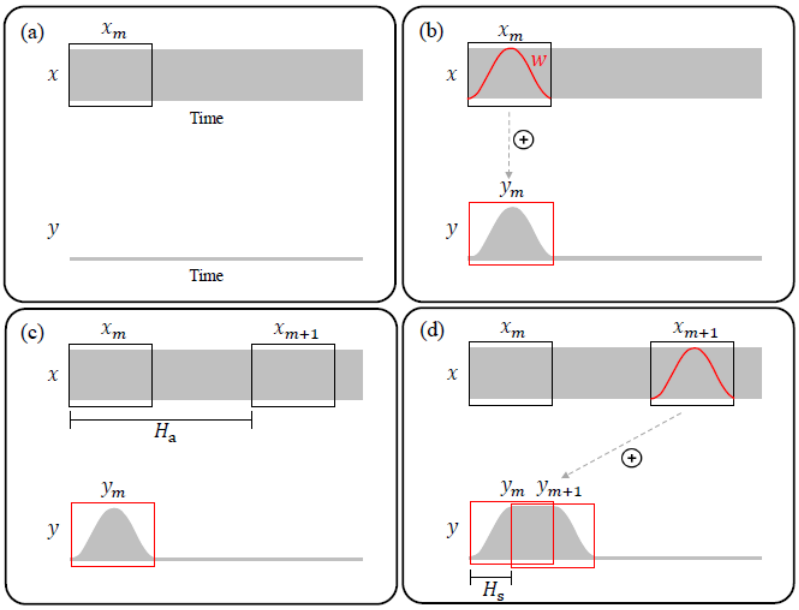

Figura 1: Representacion grafica del metodo OLA.a)Muestra el segmento m-esimo de la entrada que se copiara a la salida.b)Se aplica la ventana a dicho segmento y se lo copia a la salida.c)Toma el proximo segemento a copiar que se encuentra a una distancia H_a del anterior.d)Le aplica la ventana al segmento m+1-esimo y lo copia a su ubicacion correspondiente a la salida, que esta a una distancia $H_s$del bloque de sintesis anterior.

2.1.1. Parametros utilizados

El algoritmo necesita recibir tres parametros (ademas del vector con la senal de entrada),los cuales son:

- La ventana que se desea apliccarle a cada uno de los segmentos del input
- La funcion de mappeo definida en (2)
- El factor de overlap.

Los primeros dos parametros ya se presentaron brevemente.El factor de overlap es un numero que toma valores entre 0 y 1(no inclusive).Este factor decide la separacion H_s entre dos bloques de la entrada consecutivos como se muestra en la figura (1).La separacion H_s entre bloques de sintesis esta dada por la reacion:

$$H_s = (1 - OF).largo(Ventana)$$

[1]Algunos ejemplos notables son SOLA, PSOLA y WSOLA(el cual se analiza en este informe)

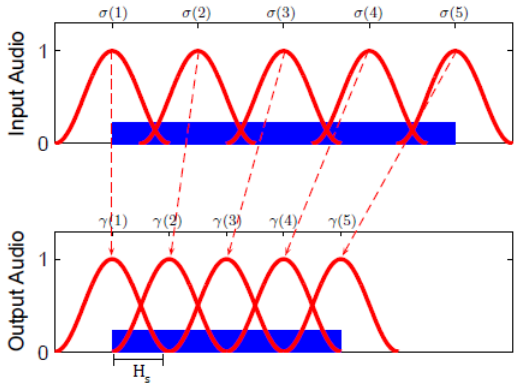

Figura 2: Posicion de los bloques de analisis y sintesis dados por σ y γ

Donde OF es el overlap factor.Como se puede ver de la relacion anterior, cuando OF toma el valor extemo $OF = 0$ entonces los bloques de sintesis estan separados por la misma distancia que el largo de la ventana,lo que quiere decir que no hay superposicion entre bloques de sintesis.En el caso extremo en el que $OF = 1$ entonces $H_s = 0$ por lo que no habria separacion entre bloques de sintesis sino que habria una superposicion del 100 % y estarian todos centrados en la misma posicion a la salida.

2.1.2. Algoritmo

Una vez definidos los parametros el metodo consiste en formar un vector con las posiciones en las cuales estan centrados cada bloque de sintesis a la salida.Esto es trivial ya que se conoce la separacion entre bloques de sintesis y que $\tau(0) = 0$.Se denotara dicho vector con la letra γ.Entonces se tiene que:

- $\gamma(1) = 0$, El primer bloque de sintesis esta centrado en 0, dado que $\tau(0) = 0$
- $\gamma(k) = \gamma(k-1) + H_s$
- $largo(\gamma) = \left\lceil \frac{largo(output)}{H_s} \right\rceil$

Una vez que se tienen los valores de este vector se puede utilizar conjuntamente con la funcion de mappeo τ para obtener las posiciones en las que estan centrados cada bloque de analisis necesario para obtener los bloques de sintesis dados por γ. Se denotara el vector con las posiciones en las que estan centrados los bloques de analisis con la letra σ.Para este vector se tiene:

- $\sigma(1) = 0$, Este caso es igual que para γ debido a que $\tau(0) = 0$
- $\sigma(k) = \tau^{-1}(\gamma(k))$, donde τ^{-1}es la funcion inversa de la time stretch function.
- $largo(\sigma) = largo(\gamma)$

El rol de estos vectores queda mucho mas claro con una representacion grafica, por lo que aqui se presenta un esquema de como es la relacion entre estos dos vectores.

Los rectangulos azules en el input y en el ouput simbolizan las muestras de audio.En el caso anterior se muestra que el rectangulo azul en el output es de menor extension que el del input, lo que significa que se esta usando el algoritmo OLA para comprimir en el tiempo el input.Sin embargo, la figura (2) se presenta para mostrar la funcion de los vectores σ y γ, la cual es la misma tanto para comprimir como para extender en el tiempo una senal.

Finalemente el audio a la slaida esta dado por la siguiente relacion matematica:

$$output = \frac{\sum_{k=1}^{largo(\sigma)} \omega(n - \gamma(k)) input(n - \gamma(k) + \sigma(k))}{A(n)} \tag{3}$$

ω hace referencia a la funcion ventanna que se recibe como parametro.La funcion $A(n)$ que se encuentra dividiendo abajao se utiliza para normalizar la salida ya que la aplicacion de ventanas puede causar distorsiones de amplitud no deseadas en la senal de salida.Para obtener la funcion $A(n)$ que normaliza el valor de $y(n)$ se parte de la base de que en un caso ideal se desea que la suma de las ventanas a la salida sea la funcion unitaria:

$$\sum_{k=1}^{largo(\gamma)} \omega(n - \gamma(k)) = 1$$

En este caso $A(n) = 1$, por lo que dividir por $A(n)$ no introduce ningun cambio a la salida.Sin embargo cuando la suma de la suerpoicion de ventanas a la salida es distinto de la funcion unitaria es necesario normalizar la salida para compensar distorsiones de amplitud, y la funcion de normalizacion esta dada por:

$$\sum_{k=1}^{largo(\gamma)} \omega(n - \gamma(k)) = A(n)$$

2.1.3. Ventajas y desventajas

La principal ventaja de OLA es que es el algoritmo mas simple de todos los implementados tanto conceptualmente como computacionalmente, por lo que es posible realizar sintesis de notas musicales de manera muy rapida si se utiliza OLA. Sin embargo, el audio obtenido es de una calidad muy baja.Para sintesis de instrumentos con contenido armonico importante(por ejemplo el violin) la distorsion introducida tapa casi por completo el sonido original del instrumento, ademas para extensiones o compresiones de tiempo grandes(por lo menos duplicar la longitud o acortarla a la mitad) la distorsion para cualquier instrumento es apreciable.

Esto se debe a que OLA siplemente copia un bloque de analisis de la entrada y lo pega a la salida, esta mecanismo simple no tiene en cuenta factores importantes para la preservacion del contenido armonico, como la continuidad de la fase.

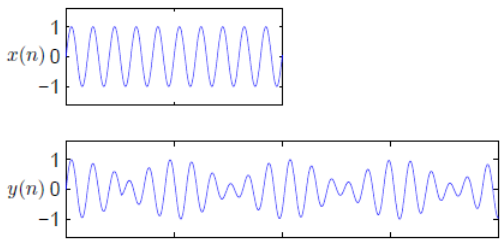

Figura 3: Distorsion armonica introducida por OLA(extendiendo la duracion de la entrada el doble)

A pesar de la pesima calidad de OLA para preservar el contenido armonico de un audio, el algoritmo puede ser sumamente util para modificar las duraciones de muestras tomadas de instrumentos percusivos(tambien referidos como 'no armonicos').Esto se debe a que en los instrmentos percusivos toma mucha mayor importancia el ritmo y las caracteristicas temporales del sonido que su contenido armonico(la introduccion de armonicos a causa de discontinuidades no es tan perceptible como para los instrumentos con armonicos bien definidos).

2.2. Waveform Synchronous Overlap and Add(WSOLA)

Este algoritmo pertence a la familia de algoritmos que son una derivacion de OLA. En principio el algoritmo sigue exactamente el mismo procedimiento que OLA.La unica diferencia entre los dos es que para WSOLA se agrega un parametro mas definido como 'tolerancia'.

Se menciono en la subseccion anterior, como el problema principal de OLA es que introduce discontinuidades en la fase a la salida que generan nuevos armonicos y distorsian notablemente el sonido resultante.Como solucion a este problema, el parametro adicional de WSOLA permite tomar cada bloque de analisis en un rango posible de ubicaciones a la entrada dado por:

$$\sigma^{'}(k) = \sigma(k) \pm \triangle_k \tag{4}$$

σ tiene el mismo significado aqui que el que se le dio en la subseccion anterior. El $\triangle_k$ es un numero que pertenece al rango de valores comprendido entre $[-\triangle_{max}, \triangle_{max}]$ donde $\triangle_{max}$ es el nuevo parametro llamado 'tolerancia' que se menciono previamente.

Lo que se esta realizando es permitir centrar el bloque de analisis en un rango de posiciones cercanas a la dada por $\sigma(k)$ con el fin de preservar lo mejor posible las caracteristicas armonicas de la senal original.El unico problema restante es como conseguir el valor de $\triangle_k$ optimo que minimice las discontinuidades a la salida. Para ello se define como 'progresion natural' de un blolque de analisis a aquel bloque que colocado a una distancia H_s del bloque de sintesis anterior no presente ninguna discontinuidad de fase.Esto se cumple siempre para el bloque de analisis cuyo centro esta dado por:

$$\sigma_{natural}(k) = \sigma^{'}(k-1) + H_s$$

Por lo que $\triangle_k$ debe ser tal que el bloque de analisis k-esimo tomado sea lo mas similar posible a la progresion natural del bloque de analisis anterior.

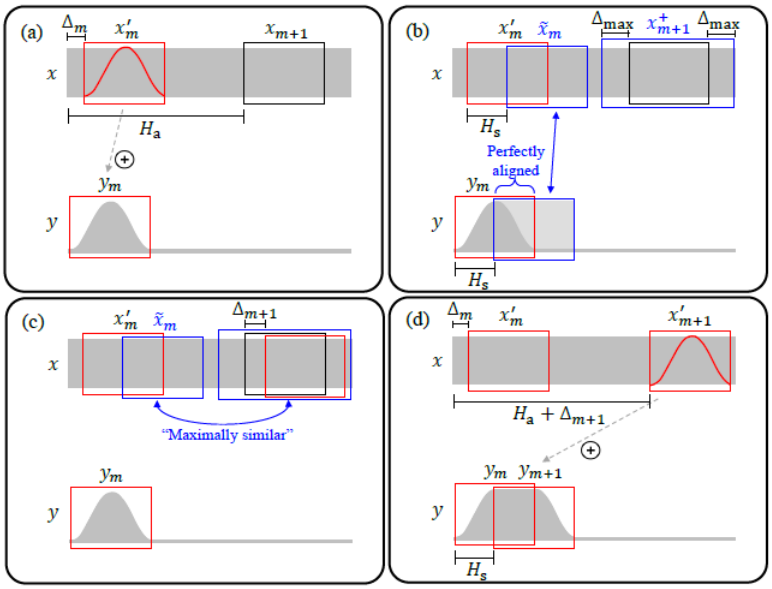

Figura 4: Grafica del metodo WSOLA.a)Se muestra el bloque de analisis m-esimo ya elegido y desplazado un $\triangle_m$de su centro original dado por σ.b)El bloque indicado por $\widetilde{x}_m$ es la 'progresion natural' de $x^{'}_m$,debe elegirse un $x^{'}_{m+1}$que se encuentre dentro del rectangulo azul de la derecha.c)Se encunetra el bloque de analisis mas similar a $\widetilde{x}_m$ desplazandose un $\triangle_{m+1}$ a la derecha de la posicion dada por $\sigma(m+1)$.d)Se toma dicho bloque de analisis que es maximamente similar a la progresion natural y se lo copia a una distancia de $H_s$respecto del bloque de sintesis anterior.

2.2.1. Calculo del $\triangle_k$ optimo

Previamente se menciono que el $\triangle_k$ optimo es aquel que maximiza la similitud entre el bloque de analisis actual y la progresion natural del ultimo bloque de analisis elegido. Para encontrar dicho bloque de analisis, realizamos la correlacion entre el bloque correspondiente a la progresion natural y el rectangulo azul de la derecha presentado en la figura(4(b)).Donde la correlacion entre dos bloques esta defindida como:

$$correlacion\{x,y\}(n) = \sum_{k=0}^{largo(ventana)} x(k+n)y(k) \tag{5}$$

La correlacion es maxima donde la suma de los productos es maxima,lo cual ocurre cuando los dos bloques son similares, entonces el $\triangle_k$ optimo se obtiene como:

$$\triangle_{k_{optimo}} = n/correlacion\{x,y\}(n) = maximo(correlacion\{x,y\}(n))$$

2.2.2. Ventajas y desventajas

WSOLA soluciona considerablemente bien el problema de la distorsion armonica que se encuentra en OLA.El algoritmo sigue siendo relativamente rapido, aunque menos que OLA ya que se debe realizar una correlacion entre dos bloques por cada bloque de sintesis en la salida. Sin embargo, el problema principal de WSOLA es que selecciona los bloques de analisis con tal de minimizar la discontinuidades de fase de la frecuencia fundamental del audio.Para instrumentos compuestos por varias frecuencias o efectos sutiles de modulacion de amplitud(como el violin o la flauta) WSOLA solo preserva la continuidad de fase para una frecuencia por lo que se distorsiona el timbre del instrumento.

2.3. Phase Vocoder

El ultimo algoritmo implementado fue el phase vocoder. Aunque el funcionamiento del phase vocoder es similar a OLA y WSOLA en que descompone a la senal de entrada en bloques de analisis por medio de la aplicacion de ventanas y luego coloca la suma de los bloques de sintesis a la salida, este algoritmo es muy distinto a los anteriores conceptualmente.

En primer lugar, el phase vocoder es uno de los algoritmos de TSM que trabaja en el dominio de la frecuencia.Como se vio previamente, el principal problema de OLA y WSOLA es su desempeno para instrumentos armonicos y con timbres complejos debido a la distorsion armonica que generan. El phase vocoder busca minimizar eso buscando que se mantenga la continuidad de fase a la salida para todos los armonicos que componen a la senal.

El algoritmo consiste en aplicarle la FFT(Fast Fourier Transform) a cada bloque de analisis que se toma del input. El resultado que se obtiene es una STFT(Short Time Fourier Transform) de la entrada, donde cada intervalo de tiempo es un bloque de analisis.Luego,se ajustan las fases de los espectros obtenidos de forma tal que al superponer la suma de los espectros a la salida no haya discontinuidades en la fase.Finalmente se aplica la IFFT a cada uno de los espectros modificados y se obtienen los bloques de sintesis los cuales son multiplicados nuevamente por una ventana antes de ser superpuestos a la salida.

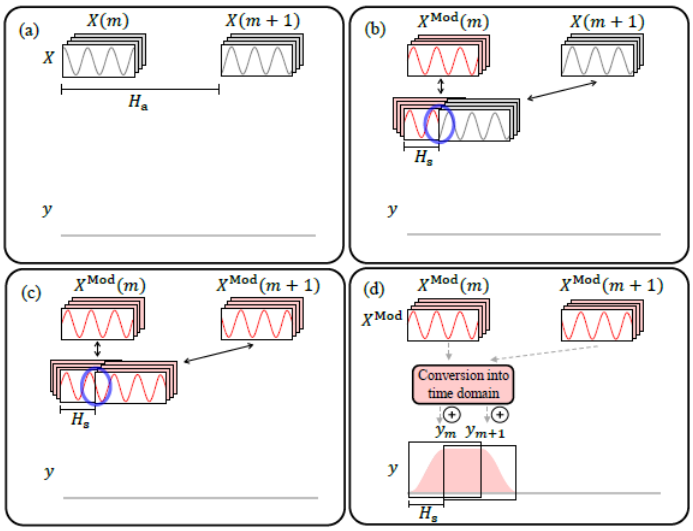

Figura 5: Esquema del funcionamiento del Phase Vocoder.a)Se tienen dos arreglos uno es $X_m$el cual se obtiene de aplicarle la fft al bloque de analisis x_my el otro es X_{m+1}el cual se obtiene de aplicarle la fft al bloque de analisis que le sigue a x_m.b)Al superponer los espectros en las ubicaciones de los bloques de sintesis se presentan discontinuidades.c)Se modifican los espectros de forma tal que ya no haya discontinuidades al superponerse a la salida.d)Se antitransforman los espectros mediante la IFFT y se obtienen los bloques de sintesis correspondientes a la salida.