DISEÑO Y DESARROLLO DE UNA APLICACIÓN DE AUDIO Y UNA APLICACIÓN DE RECONOCIMIENTO DE GESTOS PARA SER USADAS EN MUSICO TERAPIA DE EDUCACIÓN ESPECIAL.

UNIVERSIDAD DE SAN BUENAVENTURA SECCIONAL MEDELLÍN

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SONIDO

MEDELLIN

2015-07-14

DISEÑO Y DESARROLLO DE UNA APLICACIÓN DE AUDIO Y UNA APLICACIÓN DE RECONOCIMIENTO DE GESTOS PARA SER USADAS EN MUSICO TERAPIA DE EDUCACIÓN ESPECIAL.

MAURICIO SUAZA URIBE

PROYECTO PRESENTADO PARA OPTAR AL TITULO DE INGENIERO DE SONIDO

ASESOR

ANTONIO ESCAMILA

MSc IN SOUND AND MUSIC COMPUTING

UNIVERSIDAD DE SAN BUENAVENTURA SECCIONAL MEDELLÍN

FACULTAD DE INGENIERÍAS

INGENIERÍA DE SONIDO

MEDELLIN

2014

CONTENIDO

1. PLANTEAMIENTO DEL PROBLEMA.....6

2. JUSTIFICACIÓN.....7

3. OBJETIVO GENERAL.....11

4. OBJETIVOS ESPECÍFICOS.....11

5. MARCO REFERENCIAL.....12

6. ESTADO DEL ARTE……20

7. METODOLOGÍA……...…… 25

8. RESULTADOS……..40

9. CONCLUSIONES.....48

10. REFERENCIAS BIBLIOGRAFICAS…….……..…….....50

INDICE DE FIGURAS

FIGURA 1 Sensor Kinect 12

FIGURA 2 Modelo de articulaciones del cuerpo humano 14

FIGURA 3 Rastreo Esquelético..... 14

FIGURA 4 Secuencia de datos para analizar..... 15

FIGURA 5 Alineamiento entre Trazos..... 16

FIGURA 6 Condiciones Alineamiento temporal dinámico..... 18

FIGURA 7 Theremin 21

FIGURA 8 Diagrama de bloques ejercicio 1 27

FIGURA 9 Diagrama de bloques ejercicio 2 28

FIGURA 10 Diagrama de bloques delay espectral..... 29

FIGURA 11 Interfaz de usuario delay espectral..... 30

FIGURA 12 Gráfica gestos 31

FIGURA 13 Comparación entre gestos realizados a distintas velocidades..... 33

FIGURA 14 Rotación de un gesto 34

FIGURA 15 Diagrama de bloques ejercicio 3 35

FIGURA 16 Interfaz gráfica ejercicio 1 36

FIGURA 17 Interfaz gráfica ejercicio 2 37

FIGURA 18 Estudiante grupo 1 interactuando con la aplicación..... 38

FIGURA 19 Estudiante grupo 2 interactuando con la aplicación..... 39

FIGURA 20 Posición ideal para el rastreo de extremidades 40

FIGURA 21 Posiciones que generan errores de detección 40

FIGURA 22 Señal antes y después de filtrar 41

FIGURA 23 Señal antes y después de filtrar 42

FIGURA 24 Resultado de la FFT en Pure data..... 43

FIGURA 25 Resultado de la FFT en Pure data..... 44

FIGURA 26 FFT después de aplicar ventana Hanning 45

FIGURA 27 Normalización posterior a la FFT..... 46

1. PLANTEAMIENTO DEL PROBLEMA

Según el DANE (Departamento Administrativo Nacional de Estadística) en abril de 2008 en Medellín se contaba con 74.134[1] personas discapacitadas entre mujeres, hombres y niños, de estas personas el 44.1% presentaban limitaciones físicas, el 17.26% auditivas, el 43,37% visual, el 13,03% dificultades para hablar, el 12,09% para atender o aprender y el 9.86% otros problemas mentales o emocionales. Según la encuesta de calidad de vida realizada en 2012 por la alcaldía de Medellín el 83% de las personas con discapacidad viven en estratos socio económicos bajos, por lo cual muchos no tienen acceso a instituciones de educación especial o a terapias en clínicas de rehabilitación. En este proyecto se describe el proceso de diseño y desarrollo de una aplicación basada en un sensor de profundidad Kinect, la cual a través de la implementación del reconocimiento de gestos, permite a los usuarios controlar una aplicación de audio, cuyo fin es su uso en músico terapias para educación especial.

Con este proyecto, se desea investigar sobre el desarrollo e implementación de software interactivo para entornos de educación especial. Estudiar los retos que implican la creación de una herramienta, que por medio del procesamiento digital de señales, facilite y mejore las terapias, y que además permita a quienes no tienen acceso a las instituciones de educación especial, realizar las terapias desde su hogar. La investigación, se centra en el desarrollo de la aplicación interactiva, la cual permite realizar ejercicios de músico terapia, diseñados para trabajar la atención, la memorización de conceptos, y el desarrollo de la capacidad motriz tanto fina como gruesa [9]. Esta aplicación está compuesta de 3 secciones: El sensor encargado de rastrear y capturar el movimiento de las extremidades del cuerpo humano, otra parte encargada de recibir los datos provenientes del movimiento, realizar el análisis de reconocimiento de gestos y generar mensajes correspondientes a los resultados del análisis, y finalmente la parte encargada de recibir estos mensajes y generar la señal de audio correspondiente.

[1] Información tomada del estudio Identificación de las personas con discapacidad en los territorios desde el rediseño del registro (2008).

2. JUSTIFICACIÓN

La músico terapia es un proceso terapéutico-educacional para niños y adultos con discapacidades cognitivas y motrices, la cual permite impulsar su desarrollo cognitivo, perceptual, motriz, social y emocional [7]. En estas terapias los pacientes interactúan con la música de diversas maneras, y se busca establecer retos que fomenten el desarrollo del paciente en las áreas en las que presente problemáticas. La aplicación desarrollada implementa ejercicios para trabajar la atención, la memorización de conceptos, y el desarrollo de la capacidad motriz tanto fina como gruesa [9]. Permite a los usuarios realizar ejercicios tales como completar una melodía o interpretar un instrumento a través de una secuencia de movimientos. A continuación se dará una visión general de los métodos, alcances y aplicaciones de la músico terapia para personas con discapacidades cognitivas.

- Las investigaciones en el funcionamiento neurológico soportan la asociación entre la música y el desarrollo cognitivo. La música organiza silencios y sonidos en un flujo de tiempo, hace una pregunta y luego la resuelve, es usada para recordar secuencias, usa estructuras categóricas para diferenciar, reconocer patrones, y conectar sonidos con un concepto. Al Interpretar un instrumento musical se trabaja la memoria, la creatividad, la precisión, la atención y la concentración [10].

- La terapia musical es efectiva para mejorar el desempeño motriz, el ritmo estimula y organiza la respuesta muscular, ayuda significativamente a las personas con desordenes neuromusculares. La oportunidad de participar en la música motiva a las personas a intentar movimientos físicos que requieren un esfuerzo extra. Por otro lado la música suave puede incrementar la flexibilidad y disminuir las tensiones musculares hipertensas. Se trabajan los problemas de motricidad y coordinación, ya que los estudiantes deben sincronizar los movimientos. Los instrumentos se pueden configurar estratégicamente para aumentar el movimiento dependiendo de la severidad de la discapacidad [10].

- Ayuda en el desarrollo del discurso receptivo y expresivo, es una manera de facilitar el desarrollo del lenguaje, mejora la toma de decisiones, la comunicación de información y el entendimiento de patrones de lenguaje, motiva y mejora las aptitudes sociales, juega un papel que responde en una situación social, y se participa en una experiencia grupal con compañeros [9].

- La música provee de una oportunidad para expresar y experimentar una variedad de emociones. El deseo de participar y producir algo musical puede ser una motivación para controlar estallidos emocionales. El éxito que las personas con

discapacidades pueden alcanzar en la música tiene un efecto positivo en su autoestima [9].

Para el diseño de la aplicación, se tomaron en cuenta varios estudios sobre las metodologías utilizadas en músico terapia y sus efectos en el desarrollo de los pacientes. Una de las principales investigaciones en la que se basó el diseño fue la realizada por la Doctora Susan Sze de la Universidad Niagara de New York [9]. Su investigación recopila los métodos y ejercicios de músico terapia que mostraron tener resultados positivos y que se implementaron con efectividad al tratar pacientes que presentan: retardo mental, autismo, déficit de atención, problemas de aprendizaje, y algunas discapacidades físicas. Abajo se presenta una lista de estrategias que se pueden utilizar para tratar pacientes con diferentes discapacidades a través de la música.

Discapacidades de discurso y lenguaje:

- Encontrar música que sea del gusto del paciente.
- Alentarlo a participar de las actividades musicales
- Que el estudiante identifique el instrumento verbalmente antes de poder interpretarlo.
- Que aprenda palabras y articule fonemas a través del canto.
- Crear actividades en las cuales cualquier sonido vocal es aceptado como parte creativa de música improvisada.
- Incorporar sonidos vocales que sean emitidos espontáneamente, y que sean provocados al hacer música.

Discapacidades motrices y de salud:

- Ubicar un instrumento a cierta distancia para aumentar el movimiento de las extremidades.
- Tocar instrumentos de percusión.
- Que el estudiante sostenga un instrumento ayuda desarrollar el moviendo fino.
- Variar la intensidad de aplausos, saltos, giros, pisoteadas, etc. De acuerdo a la severidad de la discapacidad.
- Usar música suave y gentil para aumentar la flexibilidad, y disminuir la tensión de las contracciones musculares.

Discapacidades cognitivas:

- Usar mnemotécnicos para recordar secuencias (ejemplo la canción del

alfabeto)

- Usar estructuras categóricas para diferenciar (animales, colores).
- Conectar el sonido con un concepto (por ejemplo la vaca hace el sonido "mu").

Déficit de atención, desordenes de híper actividad, y discapacidades de aprendizaje:

- Conectar un sonido vocal particular con un movimiento del cuerpo.
- Integrar mas de un camino neuronal, usando canales multi sensoriales.
- Usar el ritmo, el tempo, y los golpes básicos de música, como un modelo para darle al estudiante un sentido del orden, la secuencia y un sentido de consistencia.

Desordenes emocionales y de comportamiento:

- Usar pequeños grupos de terapia para facilitar la socialización y las interacciones interpersonales
- Alentar al estudiante a tocar distintos instrumentos en las canciones.
- Permitirle al estudiante que se exprese individualmente mientras participa en grupo
- Usar el ritmo para fomentar la cohesión grupal
- Usar actividades en las que un miembro deba imitar los movimientos de otro
- Enseñarles a tomar turnos.
- Compartir los espacios de tiempo al tocar los instrumentos para controlar los impulsos.

Basados en los resultados de este estudio, se definieron los ejercicios y la metodología a implementarse en el sistema, así como también el tipo de discapacidades y las áreas de desarrollo (motriz, cognitivo, social…) que se desea trabajar con cada uno de los ejercicios.

Así se definió que el sistema estaría enfocado a pacientes con discapacidad cognitiva de leve a grave, los cuales pueden aprender a hablar durante la edad escolar, y se pueden beneficiar de la enseñanza de conceptos pre académicos como la familiaridad con el alfabeto y el cálculo simple. Dentro de este grupo de pacientes se tratará a quienes presenten alguna discapacidad motriz entre leve y media, quienes a través de terapia pueden llegar a desarrollar el control del movimiento fino y grueso de sus extremidades. Algunos de los síndromes que

presentan este tipo de discapacidades son el síndrome de Down, síndrome de Rett, síndrome de Dravett, síndrome de Prader Willi, autismo y síndrome epiléptico con retraso mental.

3. OBJETIVO GENERAL

Diseñar y desarrollar un sistema que a partir del reconocimiento de gestos controle una aplicación de audio que implemente 3 ejercicios usados en músico terapias de educación especial.

4. OBJETIVOS ESPECÍFICOS

- Desarrollar una aplicación de análisis para reconocimiento de gestos que se pueda implementar en tiempo real.
- Desarrollar una aplicación de audio que implemente 3 ejercicios interactivos apropiados para sesiones de músico terapia de educación especial.
- Implementación de la aplicación en una prueba piloto, supervisada por un terapeuta de educación especial.

5. MARCO REFERENCIAL

5.1 Sensor de entrada Kinect

Figura 1. Sensor Kinect.

(Fuente: http://www.microsoft.com/en-us/kinectforwindows)

El sensor Kinect es una barra horizontal conectada a una pequeña base con un eje motorizado, el dispositivo posee una cámara RGB que opera a 12 cuadros por segundo a una resolución de 1280 x 960, o a 30 cuadros por segundo en una resolución de 640*480. Posee también un arreglo de 4 micrófonos para capturar sonido, los 4 micrófonos permiten grabar sonido proveniente de una dirección específica, así como encontrar la ubicación de una fuente sonora. Posee un sensor de profundidad, el cual funciona mediante un emisor de rayos de luz infrarroja, y el receptor que lee los rayos reflejados de vuelta, estos rayos reflejados son convertidos en información sobre profundidad midiendo la distancia entre el objeto y el sensor, lo cual hace posible la captura de una imagen de profundidad. Estas características permiten al Kinect implementar reconocimiento avanzado de gestos, reconocimiento facial, y reconocimiento de voz.

El SDK de Kinect contiene las herramientas API (aplication programming interface) y ejemplos de código para simplificar el desarrollo de aplicaciones con fines comerciales, su SDK permite implementar un entendimiento profundo de las características físicas humanas, incluyendo el rastreo facial, esquelético y el reconocimiento de voz. Kinect también puede capturar el color de la escena junto con su profundidad durante un escaneo y luego reconstruir los datos en modelos de tres dimensiones.

.

5.2 Rastreo Esquelético

El rastreo esquelético es un proceso mediante el cual se detecta la silueta del cuerpo humano y se generan datos correspondientes a la posición de las articulaciones y extremidades . Este realiza la detección de 20 puntos estratégicos del cuerpo humano y entrega valores entre -1 y 1 correspondientes a su posición en los ejes X,Y y Z. X representa la posición horizontal con respecto al punto central de visión del sensor, Y la altura con respecto al rango de visión del sensor, y Z representa que tan cerca o lejos se encuentra la articulación con respecto al sensor. Para realizar el rastreo del esqueleto se utilizan algoritmos de visión artificial y procesamiento digital de imagen, a continuación se dará una visión general de cómo se realiza este proceso.

Primero se detecta el fondo de la imagen a partir de un algoritmo de umbralización (tresholding) usando el sensor de profundidad. Luego de remover el fondo a la imagen, a la silueta del cuerpo obtenida se le realiza una transformada de distancia, la cual da como resultado un mapa en el que se etiqueta cada pixel de la imagen con la distancia con respecto al punto inicial. Esta es una representación derivada de una imagen digital cuya utilidad radica en el reconocimiento de objetos de acuerdo a su localización y ubicación en la imagen. Luego se utilizan los algoritmos de Viola–Jones los cuales proveen un mecanismo para detectar objetos en tiempo real, a través de estos se detecta la cabeza y la parte superior del cuerpo. Se implementa un modelo de varas del esqueleto humano, y se representa este con un modelo de 20 articulaciones del cuerpo. Finalmente se comparan los modelos con la imagen que resulta después de realizar la transformada de distancia, y se hace el ajuste de los puntos. El rastreo del esqueleto entrega información detalladas sobre la posición de 20 articulaciones (Figura 2) del usuario en el campo de visión de la cámara[13].

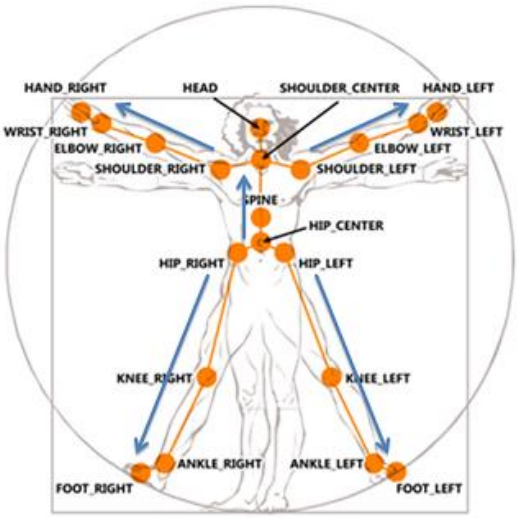


Figura 2. Modelo de 20 articulaciones del cuerpo humano utilizado por Kinect.

(Fuente: Skeletal Tracking - Web Microsoft Developer)

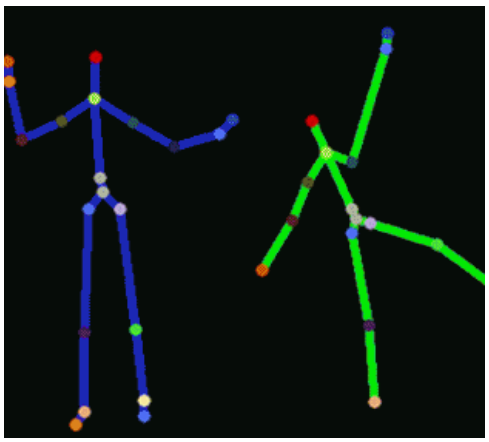


Figura 3. Resultado en pantalla luego de implementar el rastreo del esqueleto.

(Fuente: Skeletal Tracking using Microsoft Kinect, Abhishek Kar, Department of Computer Science and Engineering, IIT Kanpur)

5.3 Alineamiento temporal dinámico.

El alineamiento temporal dinámico (DTW por sus siglas en ingles), es un algoritmo que permite medir la similitud entre dos secuencias que pueden variar en el tiempo o en el espacio, es utilizado para el procesamiento digital de distintos tipos de señales con aplicaciones en áreas de telecomunicaciones, audio, o video.

Para explicar cómo funciona el alineamiento temporal dinámico, se puede tomar como ejemplo un caso de reconocimiento de símbolos de escritura, el sistema debe comparar los trazos realizados por el usuario con los almacenados y decidir cuál símbolo representa el trazo realizado por el usuario. En la figura 4, en la línea superior podemos ver los datos almacenados, y en la línea inferior una secuencia trazada por el usuario.

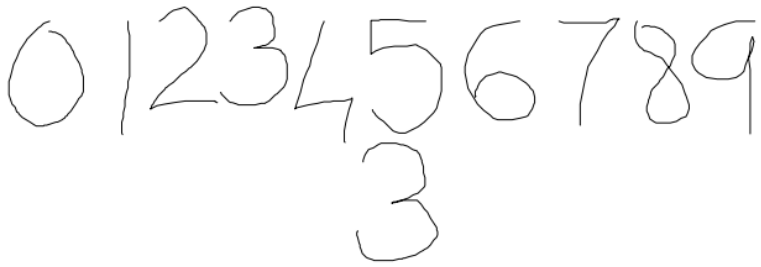


Figura 4.

(Fuente: http://lsedkigo.blogspot.com/2012/05/algoritmo-dtw-dynamic-time-warping.html)

En este ejemplo, el sistema debe clasificar esta figura ingresada por el usuario comparándola con las almacenadas. Cada figura es una secuencia de puntos en el espacio, que posee una longitud arbitraria. Ahora se compara la secuencia del digito 3 ingresada por el usuario con la secuencia almacenada. Se pueden comparar los dos trazos, midiendo la distancia euclidiana entre puntos “equivalentes”. El problema, es definir estos puntos “equivalentes”, ya que las dos secuencias pueden tener un numero diferente de puntos al haber sido realizadas a distinta velocidad, o haber seguido una trayectoria distinta. Por lo tanto, un alineamiento punto a punto no pondría en correspondencia partes similares de cada trazo. La idea de realizar el alineamiento temporal es implementar un método que permita alinear las partes similares entre los trazos (secuencias de datos), para poder medir la disimilitud entre ellos, esta disimilitud ésta representada por la suma de las distancias entre los puntos alineados.

En la figura 5 se puede observar a la izquierda el alineamiento optimo entre 2 trazos de un 3, y a la derecha el alineamiento optimo entre un trazo de un 3 y un trazo de un 4. El alineamiento se visualiza con las líneas que unen a los puntos emparejados.

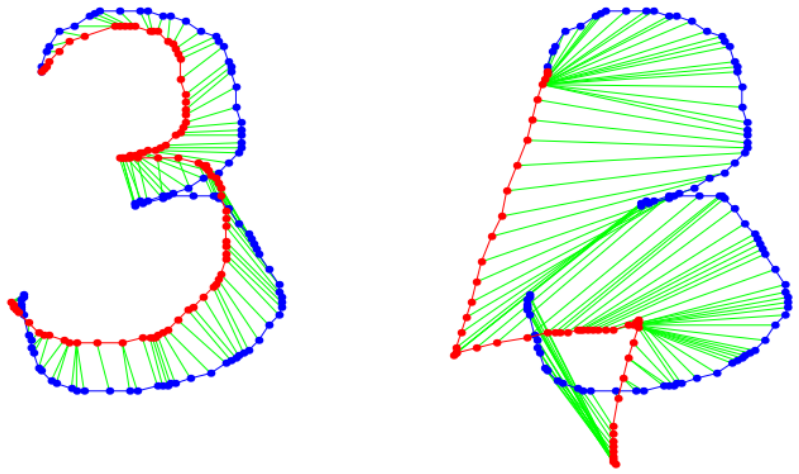


Figura 5. Izquierda. Alineamiento entre 2 trazos de un 3.

Derecha Alineamiento entre trazos de un 3 y un 4 .

(Fuente: http://lsedkigo.blogspot.com/2012/05/algoritmo-dtw-dynamic-time-warping.html)

La longitud de la línea que une los puntos es la distancia euclidiana entre los puntos. En la figura se hace evidente que la suma de longitudes entre los puntos alineados es menor al comparar las instancias de un mismo digito, que al comparar las instancias de dígitos diferentes. El alineamiento optimo es aquel que hace mínima la suma de las distancias entre los puntos alineados.

Para calcular la función de alineamiento temporal se procede de la siguiente manera. Teniendo los vectores a comparar A={a_1, a_2, ,a_3 ...a_i..a_m} y B={b_1,b_2,b_3,…a_j..b_n}, además teniendo el vector de puntos C={c(1),c(2),c(3),……c(k-1), c(k)}. En este vector cada elemento C(k) es un puntero a los elementos a comparar, C(k)= [a_i ,b_j], para cada elemento C(k) se tiene una función de coste que refleja la discrepancia entre los elementos comparados, esta función es la distancia euclidiana, d { c(k) } = d($a_{i(k)}$- $b_{j(k)}$). La función de alineamiento será aquella que minimice la función de coste total $D(c) = \sum_{k=0}^{f} d(c(k))$. Para aplicarlo se genera una matriz de 2 dimensiones de tamaño IxJ , la cual resulta al calcular la distancia de cada uno de los elementos

del vector A con respecto a los elementos del vector B. Así cada elemento de esta matriz representa la distancia entre 2 elementos del vector A y el vector B, cada camino que recorra la matriz desde (0,0) hasta (i,j) tendrá un costo asociado, y el camino con el menor costo asociado será el alineamiento optimo entre las 2 secuencias *[16]*.

Finalmente este camino de alineación se puede definir como la secuencia

$P = (p1, p2, \dots . pl), con\ pl = (n1, m1)\ \varepsilon [1, N] x [1, M], \forall\ l\ \varepsilon\ [1, L]$

Esta secuencia debe satisfacer 3 condiciones:

- Condición de frontera
  $P1 = (1,1)\ Y\ PL = (N, M)$ Para que los primeros y los últimos elementos de los vectores A y B estén alineados y así las secuencias completas estén alineadas.

- Condición de monotonía
  $n1 \leq n2 \leq \cdots \leq nl\ \ y\ m1 \leq m2 \leq \cdots \leq ml$. Con esta condición se asegura que los elementos que se sucedan en el tiempo también se sucedan al ser alineados.

- Condición de salto
  $p(i+1) - p1\ \boldsymbol{\varepsilon}\ \{(\mathbf{0}, \mathbf{1}), (\mathbf{1}, \mathbf{0}), (\mathbf{1}, \mathbf{1})\}\ \ \forall\ l\ \varepsilon\ [1, L-1]$. Esta condición indica que no se puede omitir ningún elemento de A en B y que el camino de alineamiento es continuo.

En la figura 6 se puede observar un ejemplo de caminos de alineación, y ver en que consisten cada una de estas condiciones para una secuencia X de longitud N=7 y una secuencia Y de longitud M=9. En (a) se cumplen las 3 condiciones, en (b) se viola la condición de frontera, en (c) se viola la condición de monotonía y en (d) se viola la condición de salto.

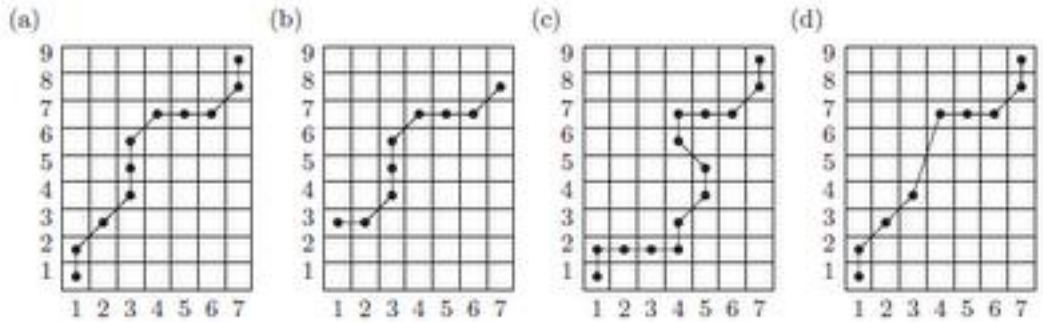


Figura 6. Condiciones para el camino de alineación temporal

(Fuente: alineamientotemporaldinamico.pdf web de la universidad autónoma de Madrid uam.es)

Este análisis de alineamiento temporal, se realiza para comparar la secuencia de datos de entrada con cada una de las secuencias almacenadas correspondientes a los gestos, y dependiendo de los valores de similitud entre las secuencias, se decide finalmente cual gesto fue ejecutado por el usuario, o si el movimiento no corresponde a ninguna de las secuencias almacenadas.

5.4 Protocolo Osc

Osc es un protocolo para la comunicación entre computadoras, y dispositivos multimedia que esta optimizado para compartir información musical en tiempo real sobre una red *[9]*. Este protocolo provee todo lo que se necesita para controlar sonido en tiempo real, de una forma flexible, fácil de implementar y permite la generación de mensajes mas complejos y precisos que el Midi.

Características principales del protocolo:

- Datos numéricos simbólicos y de alta resolución.
- Lenguaje de coincidencia de patrones para especificar múltiples receptores de un único mensaje.
- Marcas de tiempo de alta resolución.
- Mensajes “empaquetados” para aquellos eventos que deben ocurrir simultáneamente.
- Sistema de interrogación para encontrar dinámicamente las capacidades de un servidor Osc y obtener documentación.

Luego de generar los mensajes Osc estos son transmitidos a la aplicación encargada de generar el audio, a través de un protocolo llamado UDP, este protocolo permite enviar mensajes desde las aplicaciones a otros host en una red IP, sin tener que establecer canales de transmisión, su modelo de transmisión es

bastante sencillo, y permite la transmisión sin que se haya establecido previamente una conexión.

5.5 Pure Data

Pure Data es un entorno que permite la programación de aplicaciones de audio, a través de una plataforma grafica que representa el flujo de la señal. Fue desarrollado por Miller Puckette durante los años 90 para la creación de música por ordenador y obras multimedia. Aunque Puckette es el principal autor del software, es un proyecto de código abierto y tiene una gran base de desarrolladores trabajando en nuevas extensiones del programa. Pure Data tiene una base modular de código, con externos u objetos que son utilizados como bloques de construcción para programas escritos en el software. Esto hace el programa arbitrariamente extensible a través de una API pública, y alienta a los desarrolladores a añadir sus propias rutinas de audio y control, ya sea en el lenguaje de programación C o, con la ayuda de otros externos, en Python, Javascript o Ruby. Sin embargo, Pd es un lenguaje de programación en sí mismo. Unidades de código modulares y reusables, escritas nativamente en Pd, llamadas "patches" o "abstracciones", son usadas como programas independientes y compartidas libremente entre la comunidad de usuarios de Pd. Adicionalmente, Pure Data está diseñado nativamente para permitir colaboración en vivo a través de redes o de Internet, permitiendo a músicos conectados vía LAN, o incluso en distintas partes del mundo, hacer música juntos en tiempo real

6. ESTADO DEL ARTE

El enfoque tradicional para interactuar en un contexto musical y sonoro, se basa en la interpretación de instrumentos acústicos por medio de una conexión mecánica directa entre los movimientos humanos y el sonido, para controlar así el timbre y la amplificación a través de estructuras resonantes. Haciendo uso de tecnologías modernas, se puede dar la posibilidad de interactuar con la música a las personas que no tienen la capacidad de interpretar instrumentos acústicos, y además se pueden proveer nuevas formas de generar y controlar señales de audio. Avances como los algoritmos de visión artificial, sensores de profundidad, la capacidad de procesamiento de datos, los protocolos de comunicación actuales, y las plataformas de desarrollo abiertas, permiten crear interfaces humano máquina que están revolucionando la forma en la que interactuamos con los computadores.

6.1 Inicios

Desde principios del siglo XX, se han diseñado dispositivos con el objetivo de generar sonido a partir del movimiento por medio de transductores y procesamiento de datos, estos dispositivos contaban con algunos sensores básicos y aunque no disponían de la tecnología necesaria para analizar movimientos del cuerpo, ni reconocer secuencias o patrones, ya se habían ideado maneras para generar una respuesta sonora a través de una interfaz electrónica.

Un primer acercamiento al problema es el Theremin. Este fue uno de los primeros instrumentos musicales electrónicos. Inventado en 1919 por el físico y músico ruso Lev Serguéievich Termen . El diseño clásico consiste en una caja con dos antenas que se interpreta acercando y alejando la mano de cada una de las antenas correspondientes, sin llegar a tocarlas. La antena derecha suele ser recta y en vertical, y sirve para controlar la frecuencia o tono: cuanto más cerca esté la mano derecha de la misma, más agudo será el sonido producido. La antena izquierda es horizontal y con forma de bucle, y sirve para controlar el volumen: cuanto más cerca de la misma esté la mano izquierda, más baja el volumen, y viceversa. El funcionamiento del Theremin se basa en la obtención de sonidos de tono variable, utilizando una corriente alterna de frecuencia también variable. Se coloca una pequeña antena, la cual genera ondas electromagnéticas de intensidad muy débil alrededor de la misma. La aproximación de una mano, que es un conductor

eléctrico, altera el campo electromagnético que rodea a la antena, cambia su capacitancia y, por lo tanto, afecta la frecuencia de la corriente alterna generada en el instrumento.

Figura 7. Theremin.

(Fuente: http://www.indomitos.com/uploads/theremin.jpg)

6.2 IRCAM

El IRCAM (*Institut de recherche et coordination acoustique/musique*) es un instituto francés dedicado a la investigación sobre acústica y música, fundado en 1970 por Pierre Boulez, este instituto es pionero en la investigación y desarrollo de nuevas metodologías para interactuar con la música y el sonido, a través del desarrollo de sistemas de captura de movimiento, reconocimiento de gestos, e implementación de distintos tipos de sensores.

El IRCAM tiene un departamento de investigación de interacción musical en tiempo real en el cual se encuentran los investigadores Frédéric Bevilacqua y Christopher Dobrian. Estos desarrollaron un sistema con un objetivo similar al de este proyecto, el de asignar sonidos a varios gestos del cuerpo humano. Para realizar el proceso de captura del movimiento humano utilizaron un sistema de captura 3d llamado Vicon8, este sistema es utilizado usualmente para reconstruir movimientos tridimensionales en aplicaciones de animación. La precisión del sistema Vicon8 y su capacidad de capturar en 3 dimensiones les permitió implementar varios algoritmos para el análisis y reconocimiento de movimientos, y utilizarlos en la asignación de sonidos a gestos. El sistema que implementaron

tenía una primera etapa en la que se capturaba el movimiento, luego se procesaba la imagen capturada y se realizaba un reconocimiento de las trayectorias, posteriormente se hacia el análisis del movimiento y el reconocimiento de los patrones, y en la etapa final se asignaban estos patrones a sonidos [5]. Para extraer los parámetros del movimiento de las imágenes capturadas utilizaron Max/Msp. Gran parte del trabajo realizado estaba enfocado en la detección de gestos individuales a partir de una serie de datos del movimiento capturado. Para detectar los gestos se utilizaron 2 enfoques.

1) La segmentación de los gestos, usando un análisis de las curvas de aceleración. Varios parámetros pueden ser deducidos de los segmentos de los gestos, para producir eventos discretos parametrizados, que pueden ser usados junto con datos de capturas continuas del movimiento más tradicionales como la posición y la velocidad.

2) El análisis de componentes independientes. Usado para analizar patrones cortos de movimiento, que se grabaron como movimientos de referencia. Las secuencias de movimientos siguientes son comparadas con las de referencia. La similitud entre los movimientos examinados y los de referencia es cuantificada, y finalmente se realiza la asignación de los parámetros analizados a eventos de audio.

Entre otro de los proyectos actuales del IRCAM se encuentra el proyecto Legos. [18] La idea general del proyecto Legos es impulsar el desarrollo del conocimiento interdisciplinario entre los sistemas de sonido controlados a través de gestos y la neurociencia, especialmente en lo que se refiere al aprendizaje sensorio motriz. El objetivo de la investigación es tener una mayor comprensión, de los mecanismos sensorio motrices involucrados en el acoplamiento de los gestos y el sonido, y proveer metodologías eficientes para la evaluación y optimización de estos sistemas interactivos.

Las áreas que se investigan en el proyecto Legos son:

- El Aprendizaje de gestos o rehabilitación: Ejecutar un gesto determinado el cual tiene una respuesta sonora.

- El Control sonoro basado en el movimiento: Producir un sonido dado a través de la manipulación de una interfaz de gestos, como al manejar un instrumento musical digital.

- El Diseño sonoro interactivo: La tarea es manipular un objeto o una interfaz tangible que esta sonorizada.

Otro proyecto del departamento de interacción musical en tiempo real del IRCAM es el Gesture Follower [19]. Un sistema diseñado para reconocer y seguir los gestos en tiempo real y reconocer el perfil de tiempo de estos.. En la mayoría de sistemas de reconocimiento de gestos, estos son considerados unidades que deben ser reconocidas una vez completadas. Por lo tanto estos sistemas entregan los resultados en eventos de tiempo discreto, típicamente al final de cada gesto.

El sistema Gesture Follower trabaja con un paradigma distinto, el cual permite el análisis de gestos en tiempo real para lograr generar una salida continua a medida que se va ejecutando el gesto, esta salida es sensible a la velocidad con la que se ejecuta. El sistema de reconocimiento trabaja con varios parámetros que caracterizan el gesto ejecutado, estos parámetros son obtenidos al estar comparando la información de entrada con secuencias temporales almacenadas en una base de datos. Precisamente 2 tipos de información se actualizan continuamente, las cuales son estimaciones probabilísticas de:

1) La similitud del gesto en ejecución con los gestos almacenados.

2) La progresión temporal del gesto ejecutado.

El primer tipo de información permite la selección del gesto más similar en cualquier momento, y el segundo tipo de información permite estimar el índice temporal del gesto, lo cual es llamado Gesture Following. Estos datos de salida son especialmente efectivos para ambas tareas seleccionar y sincronizar procesos visuales o sonoros a gestos.

6.3 Kinect como herramienta terapéutica.

El instituto de investigación inglés Roke Manor, junto a la Universidad de South Hampton, se encuentran desarrollando actualmente un sistema de reconocimiento de gestos con Kinect para ser usado en el tratamiento de pacientes que han sufrido derrame cerebral. Los derrames cerebrales son la mayor causa de discapacidad severa en el reino unido, y la rehabilitación de los pacientes le cuesta anualmente al estado 2.5 billones de libras . La meta del dispositivo es realizar un sistema que le permita a los pacientes realizar los ejercicios de rehabilitación en su hogar, sin embargo el reto que enfrentan es de realizar un

sistema que sea lo suficientemente robusto para detectar los movimientos de la mano con la precisión requerida por los ejercicios de rehabilitación. [17]

Investigadores del Georgia Institute Of Technology en Atlanta, realizaron un sistema para reconocer lenguaje de señas a través del Kinect, para ser aplicado en juegos educacionales para niños sordos[3]. Durante su investigación compararon su sistema basado en Kinect con el sistema que se usa actualmente, el cual funciona a través de unos guantes los cuales integran acelerómetros para rastrear los movimientos de la mano. En su investigación, implementaron ambos sistemas para analizar 1000 frases del lenguaje de señas americano. Usando el Kinect el sistema dio como resultado la verificación del 51.5% y del 76.12% de las frases cuando los usuarios estaban sentados y parados respectivamente. Estas tasas son comparables con el 74.8% de verificación, arrojados por el sistema del Guante. Aunque el sistema con Kinect requiere más precisión para ser usado sentado, los resultados sugirieron que es una opción viable para el reconocimiento de lenguaje de señas.

La implementación de tecnologías multimedia en las terapias y tratamientos para personas con discapacidades cognitivas y motrices, han demostrado tener un impacto positivo en la evolución y mejoramiento de la calidad de vida de los pacientes. La Universidad de york en el reino unido ha sido pionera en investigaciones con este fin y han desarrollado varios proyectos interactivos diseñados para ser utilizados en terapias de esta índole [11], proyectos audiovisuales que invitan al paciente a intentar movimientos que usualmente les resultan difíciles, a participar en ambientes audiovisuales inmersivos en los que el sonido y las imágenes responden a la interacción entre los participantes. La implementación de estos ejercicios favorecen el desarrollo cognitivo, motriz, comunicativo, social, y emocional de los pacientes, teniendo un impacto directo en el mejoramiento de su calidad de vida. Sin embargo este es un campo bastante nuevo en el que se deben realizar más estudios para desarrollar la metodología apropiada para las terapias, y analizar los resultados de una forma cuantitativa.

7. METODOLOGIA

El desarrollo de la aplicación constó de 3 partes: El diseño de cada ejercicio basándose en las investigaciones realizadas sobre músico terapia, el desarrollo e implementación de la aplicación de reconocimiento de gestos para cada ejercicio y el desarrollo de la aplicación de audio que genera la respuesta sonora correspondiente para cada ejercicio. A continuación se explica la metodología utilizada para desarrollar cada uno de los ejercicios que implementa la aplicación.

7.1 Primer ejercicio

El diseño del primer ejercicio se basó en las técnicas utilizadas para fomentar el desarrollo motriz y el aprendizaje de conceptos a través del movimiento [9]. Este ejercicio se divide en 3 tareas: La primera es la de completar una canción sobre los números. Cada vez que el usuario realiza un movimiento similar al de golpear un instrumento de percusión, se dispara una canción que narra el número de golpes que este ha ejecutado, la meta es realizar estos golpes al ritmo de la melodía y así completar una canción que cuenta del 1 al 5. En la segunda tarea, el reto del usuario consiste seguir un ritmo interpretando el redoblante y el bombo con el movimiento de sus manos. Y en la tercera tarea, el usuario debe seguir un ritmo ejecutando el sonido de un platillo con sus manos. Con este ejercicio se desea fomentar el desarrollo de las habilidades motrices finas y gruesas, la atención necesitada para ejecutar y seguir un ritmo, el aprendizaje del concepto de los números, y la memoria. [9]

El primer paso fue configurar el entorno de desarrollo, para realizar esta aplicación se utilizó el software Processing y se instalaron las librerías oSCP5 y Simple OpenNi. La librería oSCP5 permite la creación y transmisión de mensajes OSC, y la librería OpenNi permite acceder a los datos entregados por el sensor Kinect y al kit de desarrollo para este.

Para obtener los datos de las extremidades del cuerpo humano, se implementó el rastreo esquelético provisto por el kit de desarrollo de Kinect. A este kit accedemos desde Processing por medio de la librería OpenNi. Mediante esta librería se detectan el fondo de la imagen y el usuario. Una vez el usuario es detectado, se comienza el rastreo esquelético y así podemos obtener las posiciones de las extremidades del usuario.

Una vez se está ejecutando el rastreo de las extremidades, se obtienen las posiciones de las manos derecha e izquierda y se almacenan en un vector. Los valores de posición entregados por Kinect corresponden a coordenadas 3D y están en milímetros. Estos valores se convierten a coordenadas de pixeles en 2D, es decir se pasan de un valor que representa su posición en el mundo real, a un valor que representa su posición en la pantalla y finalmente son utilizados para obtener el valor de la velocidad sobre el eje Y de cada mano, al calcularse el cambio de posición sobre este eje.

Posterior a la obtención del valor de velocidad, se implementa un filtro pasa bajos para disminuir los errores de medición en la señal entregada por Kinect. Este filtro almacena en un arreglo de longitud N, datos de velocidad y realiza el promedio aritmético de estos. N varía de acuerdo a la velocidad de procesamiento de la maquina en la que se ejecute la aplicación, para sistemas más eficientes se pueden promediar más velocidades sin generar problemas de latencia en la aplicación, para el caso de las pruebas el N escogido fue de N=3.

Luego de filtrar la señal de velocidad, se implementa un método de detección de máximos. Para esto simplemente se comparan los signos de los cambios de velocidad. Así cuando hay un cambio de signo de positivo a negativo en la pendiente de la señal de velocidad, se determina la presencia de un máximo, cuando se detecta un máximo se infiere que el usuario ha realizado con su mano un movimiento similar al de golpear un tambor. Si este movimiento se presenta por encima de un umbral de velocidad definido se procede con la generación y envío del mensaje Osc correspondiente a la detección.

Para realizar el envío de los mensajes Osc se define una dirección de red con un numero de dirección IP y un numero de puerto determinado. Este número de puerto corresponde al puerto creado en Pure Data para recibir los mensajes. Luego de configurar la dirección de red, se crea un mensaje Osc al cual se adjunta como contenido el resultado de la detección de máximos. Cuando un máximo es detectado se envía el mensaje “hit” y cuando no hay máximo se envía un mensaje vacío. A continuación se presenta el diagrama de bloques de la implementación de este ejercicio.

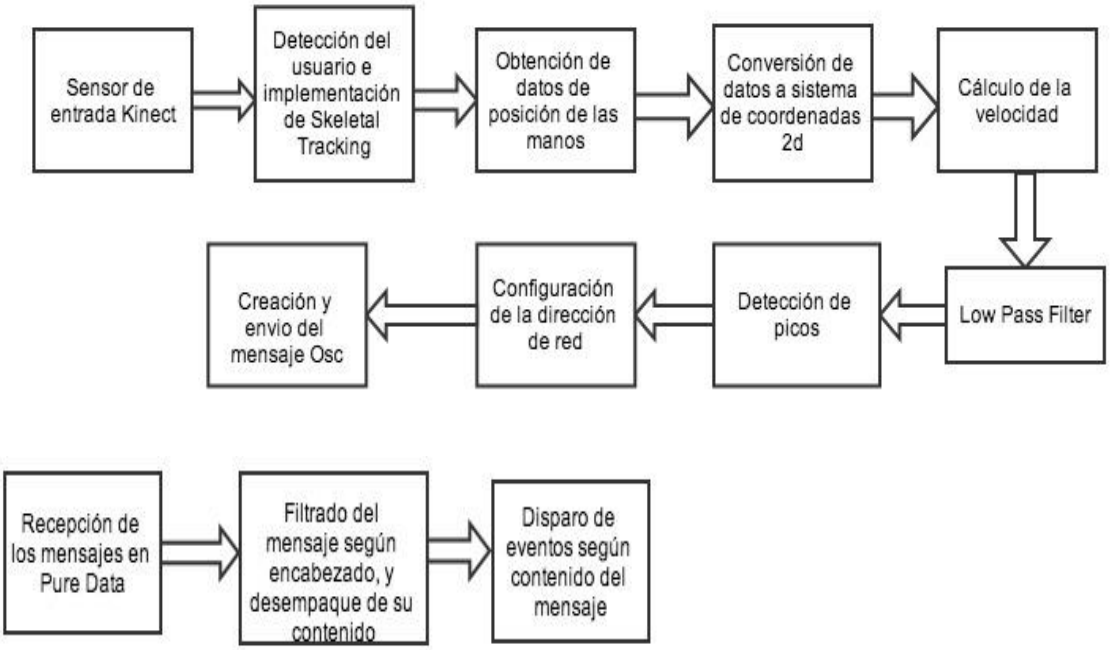


Figura 8. Diagrama de bloques del ejercicio 1

En Pure Data se registra un receptor para recibir y filtrar los mensajes que lleguen al puerto definido, este receptor se configura para recibir solamente los mensajes correspondientes a los resultados de la detección de máximos para la mano izquierda y derecha respectivamente. Luego se desempaca el contenido del mensaje y dependiendo del contenido del mensaje se dispara una acción definida. En el caso de este ejercicio los mensajes disparan la reproducción de samples de sonidos de percusión.

7.2 Segundo ejercicio

El diseño del segundo ejercicio se basó en las técnicas utilizadas en las terapias para mejorar el desarrollo cognitivo e impulsar el desarrollo motriz [9]. Este consiste en un juego interactivo para motivar el movimiento de las manos y brazos en los pacientes. En este, cuando el paciente realiza un gesto similar al de empujar un objeto hacia adelante con la mano (Gesto Click), se crea un circulo animado en la pantalla. El usuario puede crear cuantos círculos desee, y estos interactúan y generan sonidos al colisionar.

Para el desarrollo de este ejercicio se implementó el rastreo de las manos de los usuarios por medio de OpenNi, el cual posee un método predefinido para detectar las manos presentes en la imagen entregada por Kinect y rastrear su movimiento. Por la forma en la que está estructurada la librería, para comenzar el seguimiento de la mano primero se debe detectar un gesto definido para determinar la posición en la que esta se encuentra. Así definimos el gesto de Wave (Gesto similar al movimiento que se realiza al saludar con la mano), como el gesto predefinido para detectar las manos y proceder con su seguimiento.

Al estarse ejecutando el seguimiento de las manos se procede a analizar los datos de posición almacenados para detectar el gesto click, este se detecta mediante el análisis de las posiciones de la mano con respecto al eje Z. Cada vez que el gesto es detectado se dibuja en la pantalla un circulo en la posición actual de la mano. Para cada circulo creado, se definieron varios atributos como: velocidad, aceleración, y coeficiente de fricción. Estas propiedades les permiten moverse por la pantalla e interactuar entre sí. Cuando los círculos colisionan se generan tonos correspondientes a una escala musical predefinida, el reto para el usuario consiste en generar una melodía mediante su interacción con la aplicación.

Al detectar las colisiones la aplicación genera un mensaje Osc que contiene la posición en Y en la que sucedió la colisión. Antes de enviar el valor de la posición de la colisión este es mapeado a un valor entre 0 y 31, ya que luego este número es utilizado como índice para disparar notas Midi contenidas en una tabla de 32 posiciones en Pure Data. A continuación se muestra el diagrama de bloques de esta aplicación.

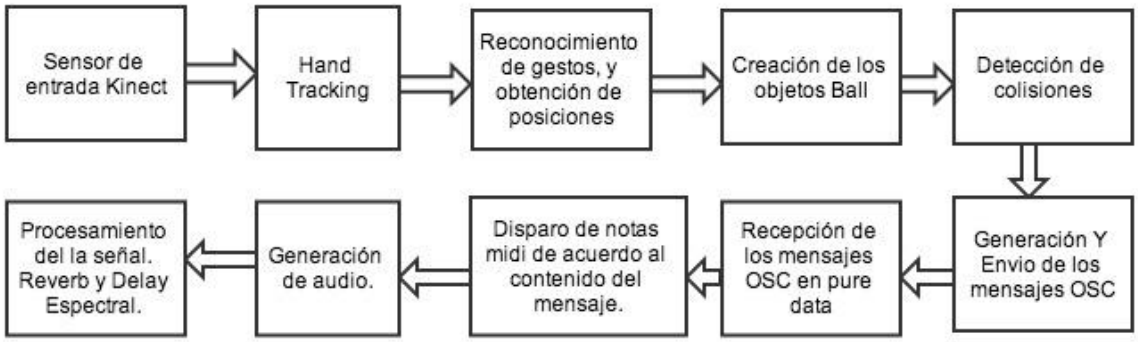


Figura 9. Diagrama de bloques del ejercicio 2.

Una vez se ha desempacado el contenido del mensaje Osc, este se utiliza como índice para leer una tabla, la cual contiene las notas Midi de una escala predefinida. Estas notas Midi se usan para disparar los sonidos de un sintetizador. El sintetizador diseñado para esta aplicación consta de 2 osciladores con selector de onda, envolventes de amplitud, un filtro pasa bajas, una envolvente de filtro y 2 efectos. Los efectos, constan de una reverberación para la cual se utilizó el módulo freeverb~ provisto por Pure Data extended, y un delay espectral, el cual fue diseñado específicamente para esta aplicación, con el fin de darle características especiales y dinámicas a su sonido.

El delay espectral, realiza una transformada rápida de Fourier a la señal y genera tiempos de retraso distintos para cada banda de frecuencia. Para su desarrollo primero se define el tamaño de los bloques de código (número de muestras de la señal que procesara cada ciclo la FFT), para este caso se utilizó un tamaño de 512 muestras, esto permite analizar frecuencias desde 86Hz en las señales. Luego se pasa la señal de entrada a través de un ventana hanning de 512 muestras, esta es una técnica que se utiliza para que la energía de la señal no quede distribuida muy ampliamente en las bandas consecutivas al realizar la transformada de Fourier. Luego de pasar por la ventana se realiza la FFT, la salida de la transformada es filtrada para tener la opción de retrasar solo cierto rango de frecuencias. Posteriormente esta señal se escribe en un buffer y este buffer es leído con un tiempo distinto de retraso para cada banda de frecuencia. Este tiempo de retraso es designado por una tabla de 512 muestras, esta tabla se lee en sincronización con los bloques procesados para que los retrasos que contiene correspondan a las información por banda resultado de la FFT. La salida de la lectura del buffer entra a un objeto que reconstruye la señal. Luego esta salida es normalizada, ya que los valores de amplitud por frecuencia se aumentan al realizarse la FFT, y finalmente se reproducen usando un ventana hanning como envolvente. A continuación se muestra el diagrama de bloques del procesamiento de señal realizado por el delay espectral.

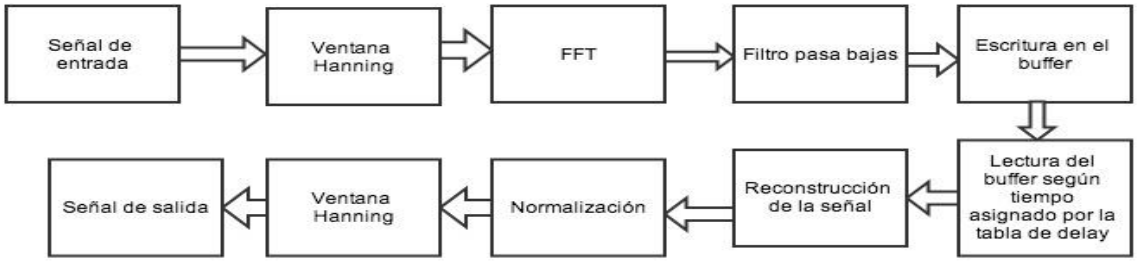


Figura 10. Diagrama de bloques del delay espectral.

En la figura 11 podemos ver la interfaz de usuario del delay espectral, la tabla $0-delay permite definir los retrasos por banda de frecuencia, la tabla $0-feedback permite definir la retro alimentación por banda de frecuencia, y la tabla $0-gain permite controlar la ganancia del resultado de la FFT.

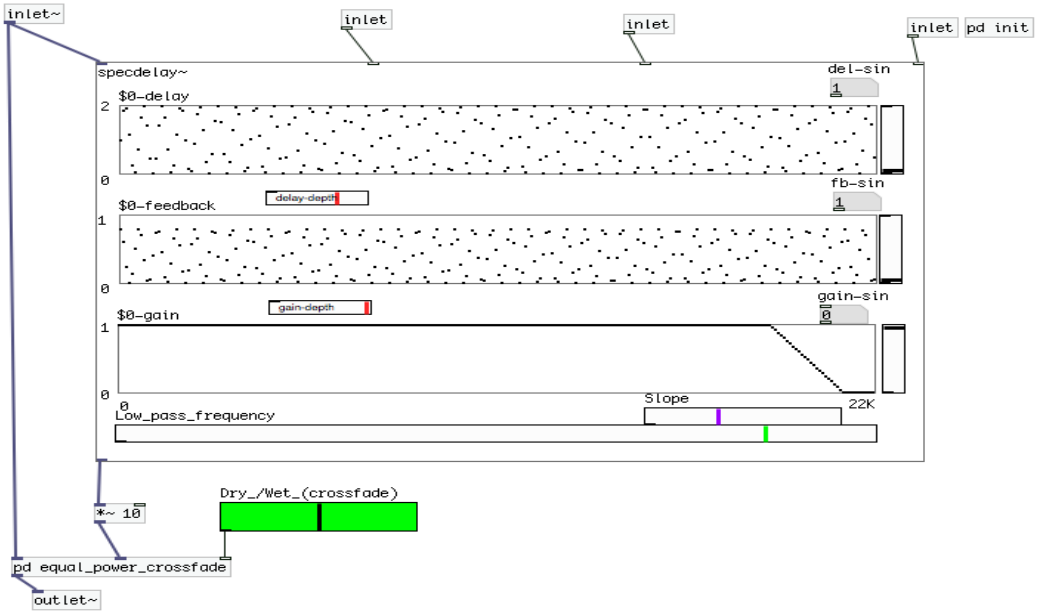


Figura 11. Interfaz de usuario del delay espectral.

7.3 Tercer ejercicio

El tercer ejercicio fue diseñado basado en las técnicas para fomentar el desarrollo cognitivo y el aprendizaje de conceptos mediante secuencias de movimientos[9]. En este ejercicio el usuario debe trazar en la pantalla con su mano las figuras correspondientes al círculo, el triángulo, y el rectángulo. Cuando el sistema detecta las figuras, se dispara una canción correspondiente a estas, el reto del usuario es lograr trazar la figura con precisión y completar la canción.

Para el desarrollo de este ejercicio se implementó el método de rastreo de manos de la misma manera que para el ejercicio 2. Las posiciones de las manos se almacenan en un mapa el cual consta de un arreglo de vectores asignados a números de identificación para cada mano. Cada frame este mapa se actualiza con la posición actual de las mano, y se compara con unos arreglos de vectores en los que están almacenadas las posiciones representativas para cada figura geométrica a reconocer. El resultado de esta comparación define si el usuario ha

trazado o no la figura geométrica correspondiente.

Para comparar la correspondencia entre el arreglo de posiciones del usuario, y el arreglo almacenado que contiene las posiciones representativas para cada figura, se utilizó el algoritmo $1 Stroke Recognizer, este fue desarrollado por investigadores de la universidad de Washington, para ayudar a los programadores a implementar interfaces de reconocimiento de gestos de forma sencilla [15], ya que los métodos de comparación de patrones requieren de un fundamente teórico bastante fuerte en conceptos como modelos ocultos de markov, redes neuronales, clasificadores estadísticos, o programación dinámica.

Para implementar el algoritmo $1 los movimientos realizados por el usuario se almacenan en una secuencia de puntos candidatos C, y se debe determinar cuál secuencia de puntos grabada previamente T, corresponde a la secuencia ingresada por el usuario. La frecuencia de muestreo de los puntos candidatos depende del sensor Kinect. Este hecho y la variabilidad de los movimientos humanos hace que la secuencia grabada T y la secuencia de candidatos C raramente se encuentren alineadas como para ser comparadas fácilmente.

Figura 12. Dos pares de gestos realizados durante 600 ms.

(Fuente: Gesture without libraries: $1 Recognizer for user interface prototypes)

En la figura 12 se muestran dos pares de gestos realizados en un tiempo de 600 ms y se marca el número de puntos correspondientes. Claramente se ve que una comparación uno a uno de los puntos es insuficiente para determinar la similitud entre los gestos. Al examinar este par de gestos los cuales corresponden a “x” y “pigtail” vemos que son de tamaños distintos y además contienen un número distinto de puntos. Esta diferencia crea un reto para el reconocimiento. También se observa que si se gira 90º el gesto “pigtail” es similar a “x”. Reflejando estos

problemas se definieron los siguientes criterios al desarrollar el reconocedor de gestos $1:

- Que no sea afectado por las variaciones del muestreo, debido a la velocidad del movimiento o a la tasa de muestreo del sensor.

- Que soporte configuraciones para rotación y escala.

- Que no requiera técnicas matemáticas avanzadas difíciles de implementar.

- Que esté escrito de forma sencilla en pocas líneas de código.

- Que sea lo suficientemente rápido para fines interactivos.

- Que permita a los desarrolladores y a los usuarios finales implementar nuevos gestos.

- Que la tasa de reconocimiento sea competente con respecto a otros algoritmos más complejos utilizados en aplicaciones de Interacción humano máquina.

El reconocedor de gestos $1 se ejecuta en 4 pasos. Primero las secuencias de puntos que se van a usar, sean secuencias de muestra o candidatos para comparar pasan inicialmente por el mismo proceso. Son re-muestreadas, rotadas, escaladas y trasladadas. Luego la distancia entre las secuencias candidatas C son comparadas contra cada una de las almacenadas T a través de una serie de ajustes angulares hasta encontrar su alineamiento angular óptimo. A continuación se explican cada uno de estos pasos:

7.3.1 Re-muestreo de la secuencia.

Los gestos en las interfaces de usuario son muestreados a una tasa determinada por el sensor y el software de entrada. Por lo tanto la velocidad del movimiento tiene un efecto claro en el número de puntos para la secuencia de un gesto.

Figura 13. Comparación entre gestos realizados a distintas velocidades.

(Fuente: Gesture without libraries: $1 Recognizer for user interface prototypes)

En la figura 13 podemos comparar 2 pares de gestos realizados a distintas velocidades. A la izquierda se pueden ver los gestos del signo de pregunta y el triángulo realizados a una velocidad lenta, estos cuentan con 60 puntos y 86 puntos respectivamente. En la derecha se pueden observar los mismo gestos realizados a una velocidad rápida, estos cuentan con 44 y 47 puntos respectivamente.

Para hacer que las secuencias de puntos sean comparables directamente, inclusive a distintas velocidades de movimiento. Se re muestrean los gestos para que la secuencia original definida por M puntos sea ahora definida por N puntos espaciados equidistantemente. Usar una N muy pequeña causa una pérdida de precisión, mientras que usar una N muy grande añade tiempo a la comparación de las secuencias.

Para re-muestrear primero se calcula el tamaño total M de la secuencia de puntos. Luego se divide este tamaño por (N-1), obteniendo así el tamaño de cada incremento i entre los N nuevos puntos. Luego la secuencia se procesa paso a paso de manera que cuando la distancia entre 2 puntos exceda el incremento, se añada un nuevo punto a través de interpolación linear. Al final de este proceso la secuencia de candidatos tiene exactamente N puntos, lo que permite medir la distancia entre cada uno de los puntos de las secuencias.

7.3.2 Rotación basada en el ángulo indicativo.

Para determinar el ángulo al que se debe rotar una secuencia de puntos para estar alineada con la de muestra, primero se haya el “ángulo indicativo” . Este se define como el ángulo formado por el centro del gesto y su primer punto. Luego se rota el gesto para que este ángulo sea 0º .

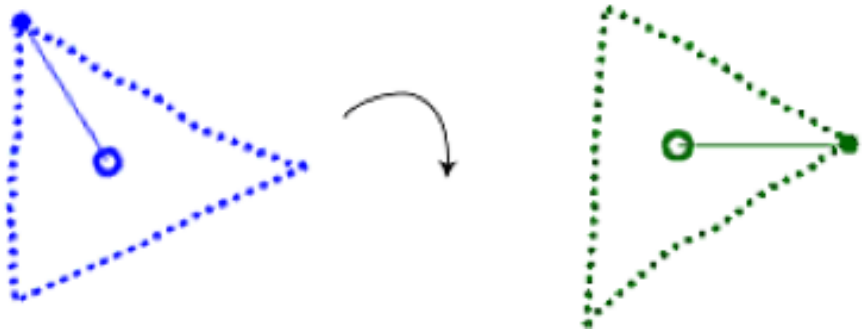


Figura 14. Rotación de un triangulo para que si su ángulo indicativo este a 0º.

(Fuente: Gesture without libraries: $1 Recognizer for user interface prototypes)

7.3.3 Escalar y trasladar.

Luego de la rotación el gesto es escalado al interior de un cuadrado de referencia. Al hacer esto se está escalando de manera no uniforme. Esto permite rotar al candidato sobre su centro, y asumir de forma segura que los cambios en los puntos emparejados entre C y T se deben solamente al ángulo de rotación y no a al tamaño de la figura. Luego de ser escalado, el gesto es trasladado a un punto de referencia. Por simplicidad el gesto es trasladado al punto de referencia (0,0)

7.3.4 Encontrar el ángulo óptimo para la mejor comparación.

En este punto todas las secuencias de candidatos y secuencias de muestra han pasado por el mismo proceso. Han sido re muestreadas, rotadas, escaladas y trasladadas. Luego de esto se pasa al proceso que se encarga de hacer el reconocimiento. Mediante la ecuación 1 cada secuencia de candidatos C (secuencia ingresada por el usuario) es comparada con cada secuencia almacenada T (secuencia de muestra de las figuras), para así encontrar la distancia promedio entre los puntos correspondientes:

$$di = \frac{\sum_{k=1}^{N} \sqrt{(C[K]x - T[K]x)^2 + (C[K]y - T[K]y)^2}}{N} \qquad (1)$$

Con esta ecuación se definen las distancias entre las secuencias C y T. La secuencia T cuya distancia sea mínima con respecto a C es el resultado del reconocimiento. Esta distancia mínima es convertida a un puntaje entre 0 y 1 usando la ecuación 2.

$$puntaje = 1 - \frac{di}{\frac{1}{2}\sqrt{S^2+S^2}} \qquad (2)$$

En esta ecuación, S corresponde al tamaño de un lado del cuadrado de referencia al cual todos los gestos son escalados en el tercer paso.

Cuando una de las figuras es detectada, se crea y se envía un mensaje Osc que contiene el nombre de la figura detectada, y se reinicia el arreglo en el que se almacenan las posiciones de la mano del usuario. Una vez el mensaje es recibido en Pure Data, se dispara una grabación de una melodía correspondiente a la figura detectada. A continuación se muestra el diagrama de bloques del funcionamiento de este ejercicio.

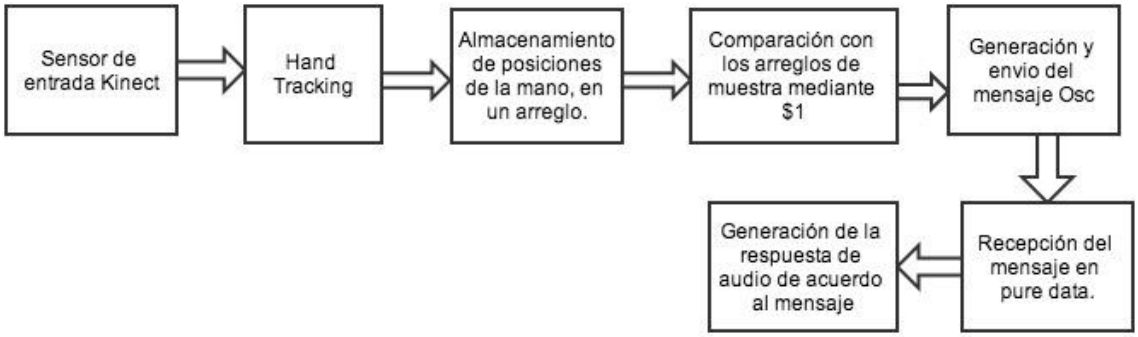


Figura 15. Diagrama de bloques del ejercicio 3.

7.4 Desarrollo de la interfaz grafica de usuario.

Posterior al desarrollo del aplicativo de reconocimiento de gestos, y del aplicativo de audio en Pure Data, se desarrolló la interfaz gráfica de usuario para cada ejercicio. Para esta tarea se utilizó Animata, un software de código abierto de animación en tiempo real, este software permite crear un esqueleto para imágenes estáticas, y producir una red de triángulos los cuales se asignan según su localización al hueso y articulación correspondiente. El movimiento de los huesos se basa en modelado físico y permite que los personajes se muevan con

facilidad, el cambio de posición de un hueso o articulación genera cambios en el resto del esqueleto dependiendo de cómo éste configurado el modelo, dándole así fluidez y naturalidad a los movimientos de los personajes. El envío de los datos de posición del usuario al modelo de Animata se realizó por medio de mensajes OSC, así se envían las posición de las extremidades del usuario y con respecto a estas posiciones se mueven las articulaciones y extremidades del modelo en Animata. En las figura 16 y 17 podemos ver los modelos desarrollados para los ejercicios 1 y 2 respectivamente.

Figura 16. Interfaz gráfica ejercicio 1.

Figura 17. Interfaz gráfica ejercicio 2.

7.5 Pruebas de campo

Las pruebas de campo de la interfaz se realizaron en la fundación Procer ubicada en la ciudad de Cali Colombia, estas fueron llevadas a cabo bajo la supervisión de la directora de la fundación María Piedad Bolaños, quien es licenciada en educación para personas con trastornos neuro-motores de la escuela de especialización de Polanco de México y lleva más de 20 años trabajando en educación de personas con discapacidades cognitivas.

Para la prueba se dividieron los estudiantes en 2 grupos, el primero era de jóvenes con retraso mental leve, los cuales en su mayoría tenían buenas habilidades motrices. El segundo grupo, estaba compuesto por jóvenes con diferentes trastornos cognitivos los cuales involucraban un retraso mental mucho más pronunciado (entre moderado y severo) y capacidades motrices disminuidas. Las pruebas se realizaron con los 3 ejercicios desarrollados de la siguiente manera:

Para el primer ejercicio los alumnos debían interpretar un ritmo con sus manos izquierda y derecha de forma libre, posteriormente debían hacerlo siguiendo un tempo indicado, y finalmente debían realizar este ritmo siguiendo una pieza musical.

El segundo ejercicio consistía en disparar las notas de una melodía en forma secuencial, de acuerdo a un gesto percutido similar al del primer ejercicio. Para realizar el ejercicio primero se reproducía la melodía como forma de entrenamiento y/o preparación. Luego los pacientes debían interpretar la melodía moviendo sus manos para disparar cada nota de forma sincronizada y al ritmo de un acompañamiento musical correspondiente.

El tercer ejercicio se trataba de trazar con la mano en el aire, la figura correspondiente a un circulo, triángulo o un rectángulo, al programa detectar la figura se dispara una melodía correspondiente a esta, y se solicitaba al usuario realizar la siguiente figura para continuar la canción.

Figura 18. Estudiante del grupo 1 interactuando con la aplicación.

Para los estudiantes del primer grupo realizar los ejercicios fue bastante fácil. Debido a sus capacidades motrices desarrolladas, podían seguir el ritmo del primer ejercicio con facilidad, además estos estudiantes también tenían buena atención y concentración, por lo que entendían la dinámica del ejercicio fácilmente. Todos se vieron muy animados al interactuar con la aplicación, y luego de practicar un tiempo eran capaces de realizar los 3 ejercicios con facilidad. En este

grupo hubo 2 estudiantes que tenían problemas motrices graves, por lo cual se les dificulto más realizar los ejercicios, a pesar de esto luego de practicar un tiempo, lograron realizar el ritmo y la melodía al tiempo indicado, mas no de trazar las figuras geométricas.

Al realizar las pruebas con el segundo grupo, se encontró un mayor interés por parte de los alumnos, sin embargo como era de esperarse fue mucho mas difícil para ellos realizar los ejercicios, involucrando un mayor esfuerzo. Para este grupo realizar el ejercicio del ritmo era más difícil debido a sus problemas motrices, sin embargo se esforzaban por realizarlo y finalmente conseguían por unos segundos ejecutar el ritmo de la canción al tiempo indicado. También presentaban una mayor dificultad para entender el objetivo del ejercicio, ya que sus capacidades de atención y concentración estaban muy poco desarrolladas, sin embargo demostraron gran empeño, motivación y esfuerzo para realizar los ejercicios.

Figura 19. Estudiante del grupo 2 interactuando con la aplicación

8. RESULTADOS

8.1 Posicionamiento óptimo para detección del usuario.

Al implementar el rastreo esquelético, se concluyó que para que el usuario y sus articulaciones sean detectados y seguidos de forma eficiente, este se debe ubicar de frente con respecto el sensor. Se encontraron varias posiciones y movimientos que generan errores de detección como cruzar las manos y ubicarse de lado con respecto al sensor. En la figura 20 se puede observar la posición ideal para la captura de los movimientos y en la figura 21 se pueden apreciar los errores de detección generados al usuario cruzar los brazos, y ubicarse de lado con respecto al sensor.

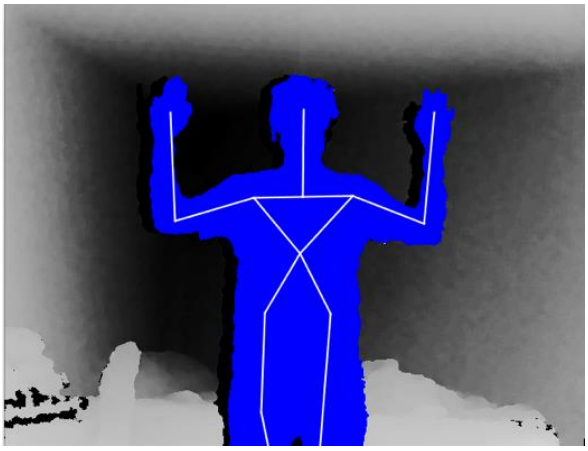


Figura 20. Posición ideal para rastreo de las extremidades

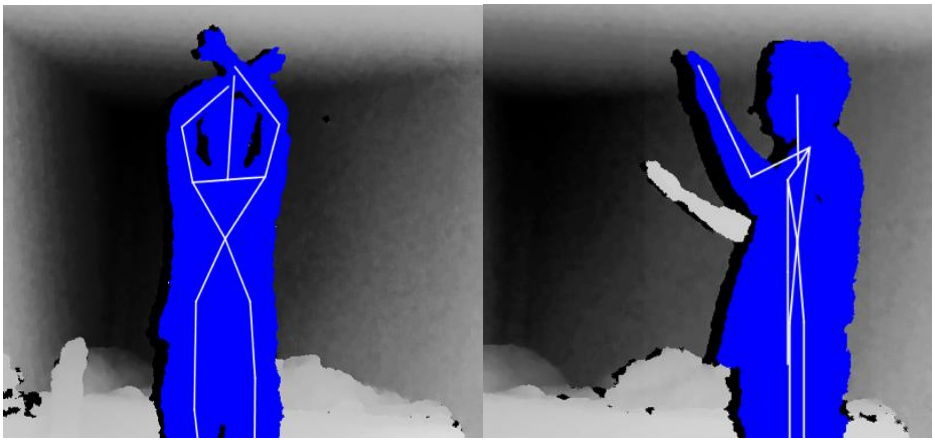


Figura 21. Posiciones que generan errores de detección

8.2 Latencia vs reducción de ruido de la señal.

Para el primer ejercicio, se desarrolló un filtro pasa bajos que promedia los valores de velocidad para reducir los errores de medición del sensor, con este método de filtrado la aplicación funciona con un tiempo de latencia bastante bajo, y la detección de máximos es eficiente. Sin embargo existe un compromiso entre la cantidad de reducción de errores de medición y la baja latencia con la que se pueden detectar los máximos. Esto se debe a que al promediar los datos de entrada, mientras más datos se tomen más tiempo se demora la señal en actualizarse. Si se promedia la señal cada 2 frames (Cuadros) se permite que la detección de máximos funcione con poca latencia, y se reduce considerablemente el ruido de la señal, aunque no se atenúan completamente los errores de medición del sensor.

La forma en la que se soluciona esto, es definir un umbral de velocidad que se debe sobrepasar para que se disparen los eventos, así los errores de medición atenuados por el filtrado no alcanzan a disparar ningún evento mientras no superen el umbral de velocidad definido. Por otro lado, lo que se puede hacer es tomar más muestras de la señal para promediar, disminuyendo así considerablemente el ruido. Sin embargo esto causa que la señal se retrase un intervalo amplio de tiempo y que se pierda la sensación de interpretación en tiempo real. A continuación las figuras 22 y 23 muestran graficas de las señales antes y después de ser promediadas, cada 2 frames y cada 5 frames respectivamente.

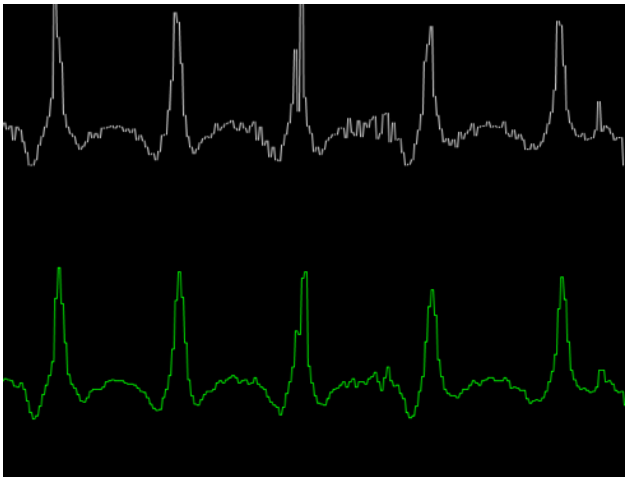


Figura 22.

En la figura 22 la señal de color blanco representa la señal directa proveniente del sensor, y la de color verde representa la señal después de realizarse el promedio cada 2 frames, como se puede observar se logra una atenuación leve del ruido de la señal y los máximos de la señal continúan en fase.

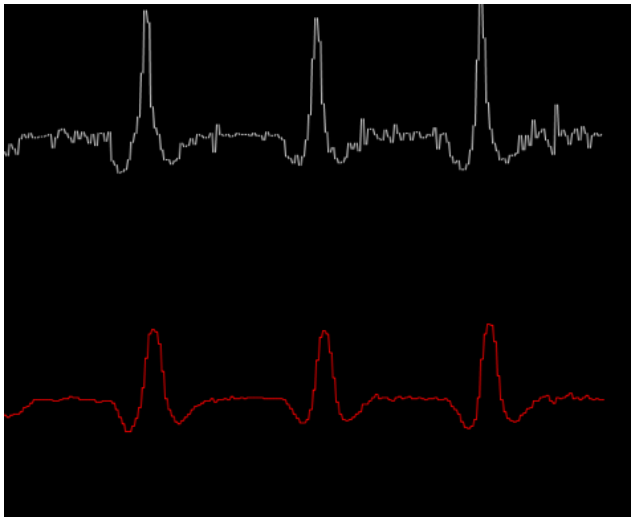


Figura 23. Señal promediada cada 5 Frames

En la figura 23 la señal de color blanco representa la señal directa proveniente del sensor, y la de color rojo representa la señal después de realizarse el promedio cada 5 frames. Como se puede observar, se logra reducir considerablemente el ruido de la señal y atenuar los errores de medición, sin embargo se retrasa la señal y los máximos están desfasados.

8.3 Ventanas y distribución de la energía de la FFT.

El delay espectral desarrollado para el segundo ejercicio permite generar retrasos diferentes por cada banda de frecuencia de la señal. Para realizar la FFT utilizada por el delay espectral, primero se debe definir el tamaño del bloque de procesamiento en Pure Data. Este tamaño representa el número de muestras que analiza la FFT cada ciclo de procesamiento, y también el mismo número de datos que se entregan correspondientes a la magnitud de la amplitud por banda de frecuencia de la señal. Si por ejemplo se escoge un bloque de procesamiento de tamaño 64, se entregan 64 datos correspondientes a la magnitud de la amplitud de la señal por banda, y estas bandas de frecuencia tendrían un ancho de 44100/64=689 Hz cada una.

En la figura 24, se muestra en consola el resultado arrojado al realizar la FFT a una señal con frecuencia de 1378 Hz. En este ejemplo el tamaño del bloque de procesamiento es de 64, y la frecuencia de la señal a analizar corresponde al segundo múltiplo de 689 Hz.

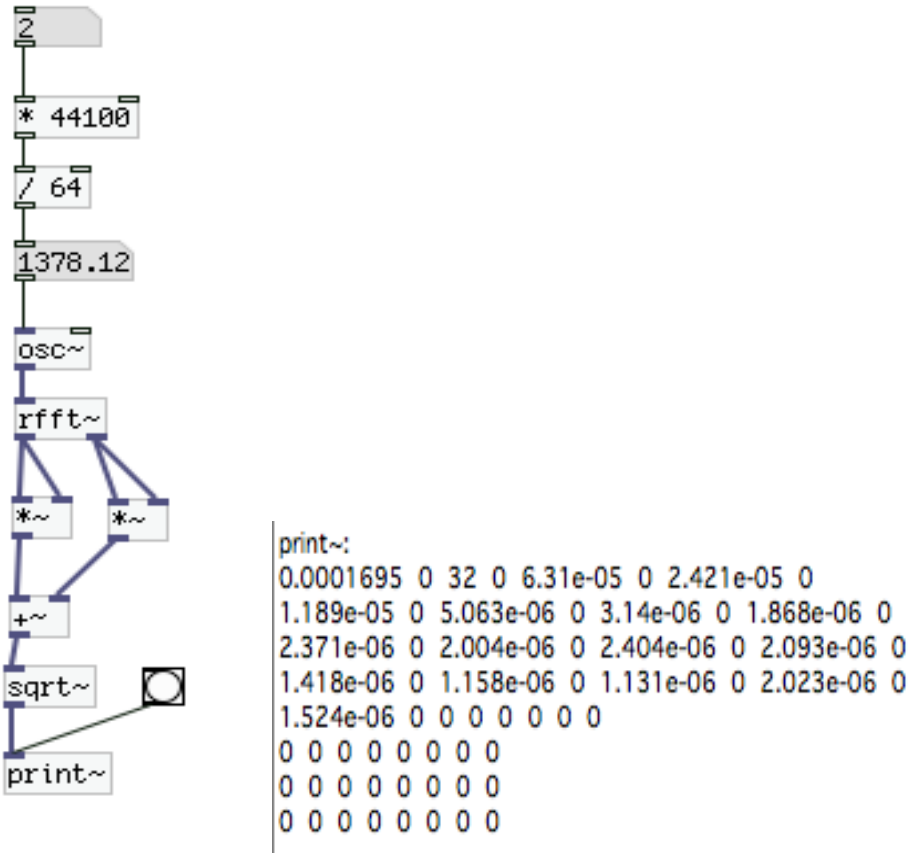


Figura 24: Resultado de la FFT en Pure Data

Como se puede ver en la impresión de la consola la energía se concentra en la segunda banda, la cual aparece con una amplitud de 32 mientras que en las otras bandas la amplitud es muy cercana a 0.

Con un tamaño de bloque de 64 muestras, se puede analizar fácilmente el espectro de una frecuencia fundamental de 689 Hz. Sin embargo en la figura 25 se ilustra el resultado de realizar el análisis para una frecuencia la cual no es múltiplo de 689hz.

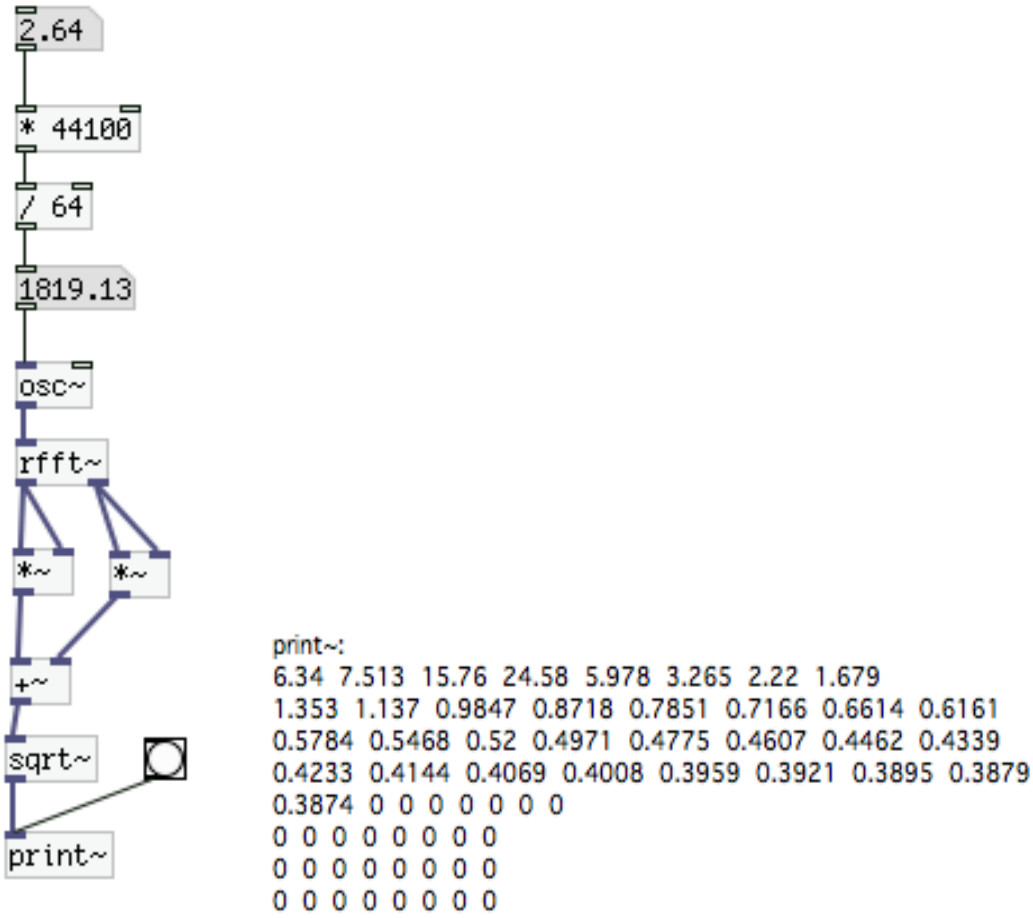
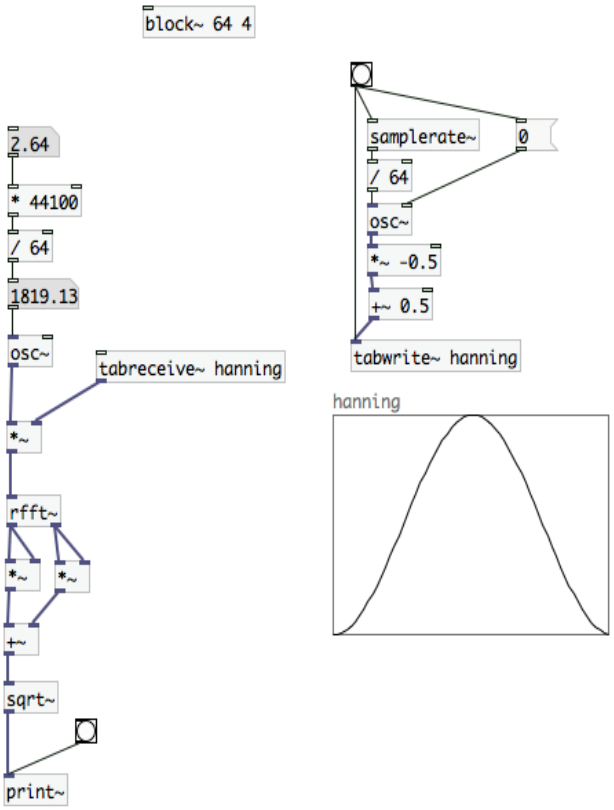


Figura 25. Resultado al realizar la FFT para una frecuencia no múltiplo de 689 Hz

Como se puede observar en la impresión de la figura 25 cuando la frecuencia no es un múltiplo exacto de la división entre la frecuencia de muestreo y el tamaño de muestras de bloque, la energía es repartida de forma muy amplia en las bandas aledañas, para resolver este problema se recurre al uso de una ventana, esta técnica permite que la energía no se reparta de una manera tan amplia en las bandas aledañas. En la figura 26 se muestra le resultado al aplicar una ventana hanning a la señal antes de realizar la FFT.




```
print~:
0.2751  1.608  12.22  14.69  3.997  0.4332  0.1375  0.06171
0.0333  0.02011  0.01321  0.009076  0.006592  0.00491  0.003774  0.002953
0.002358  0.001901  0.001565  0.001282  0.001074  0.0008879  0.0007501  0.0006233
0.000525  0.0004358  0.0003556  0.0002898  0.0002292  0.0001726  0.0001247  8.622e-0
5
6.68e-05  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
0  0  0  0  0  0  0  0
```

Figura 26. Resultado de la FFT después de aplicar la ventana Hanning con overlap.

Como se puede observar en la figura 26 al aplicar la ventana Hanning a la señal antes de realizar la transformada, la energía se reparte de una forma más precisa y menos dispersa en las bandas aledañas. Esta técnica es efectiva al momento de querer implementar un análisis de espectro más exacto y eficiente. Sin embargo se debe tener en cuenta que al reconstruir la señal esta se debe procesar de nuevo a través de la ventana.

Otro proceso que se tuvo en cuenta al realizar la FFT es el de normalización, ya que los valores que se generan de amplitud al reconstruir la señal después de realizar la transformada, se aumentan debido a las sumas que se hacen al reconstruirse la señal con la información de amplitud por cada banda. Para realizar la normalización simplemente se divide la señal de salida por el tamaño de bloque designado. En la figura 27 se muestra como aumenta el valor de la amplitud de la señal al reconstruirla después de la FFT, y como este es normalizado al dividir la señal por el tamaño del bloque.

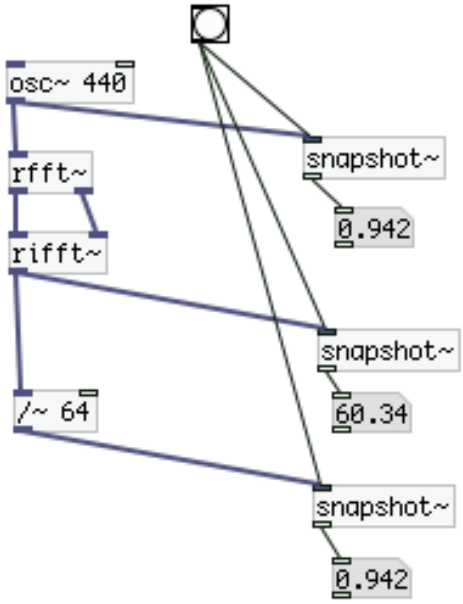


Figura 27 Normalización posterior a la FFT.

8.4 Desventajas y ventajas del algoritmo $1 stroke recognizer

El algoritmo de reconocimiento de gestos $1 implementado para el tercer ejercicio, es un comparador de plantillas geométricas. Para facilitar la comparación de los puntos emparejados, el resultado del análisis realizado por el algoritmo, no depende de las variaciones de rotación, escala, ni posición de los gestos procesados. Por lo tanto no se pueden distinguir gestos cuya identidad dependa de orientaciones, tamaño o ubicación en la pantalla. Esto afecta de manera positiva la implementación de nuestro sistema, ya que nos permite no tener que

exigir al paciente que realice los gestos con cierto tamaño, ni teniendo en cuenta cierta ubicación u orientación. Sin embargo estas características del algoritmo no permiten implementar ciertas formas geométricas, ya que con estas condiciones no se puede diferenciar entre cuadrados y rectángulos o círculos y óvalos.

Por otro lado, el algoritmo de detección de gestos compara cada iteración, cadenas de datos de longitud determinada que se actualizan constantemente, situación que genera dos tipos de problemas: En primera medida, el hecho de que la longitud del arreglo de valores de profundidad almacenados no alcance para guardar un gesto completo (figura geométrica) si el usuario realiza el movimiento a una velocidad lenta. Lo anterior implicó que exista un límite inferior en la velocidad de ejecución del gesto para que pueda ser detectado. Para ello, pensando en facilitar el desempeño de los pacientes, la aplicación muestra una animación que dibuja en pantalla la figura a la velocidad que el usuario debe ejecutarla para que el sistema de reconocimiento funcione sin problema. El segundo inconveniente, ésta relacionado con movimientos realizados por el paciente en los que la mano permanece quieta antes de ejecutar un gesto, haciendo que el conjunto de datos a comparar tenga muchos puntos en el mismo lugar espacial y no sea ideal para la detección. Para dar solución a ello, antes de validar un nuevo dato para ser usado en la detección de gestos, se verifica que esté a cierta distancia mínima respecto al último valor validado, evitando así tomar puntos muy cercanos entre si.

9. CONCLUSIONES.

Para la implementación de la tarea de reconocimiento de figuras geométricas trazadas con la mano, fue necesario entrenar el algoritmo $1 Stroke Recognizer con un conjunto representativo de gestos hechos por personas para cada figura, usando el sensor Kinect. Una característica del conjunto de entrenamiento que mejoró notablemente el comportamiento del algoritmo de detección fue la inclusión de gestos ejecutados en ambos sentidos. Sin embargo la misma naturaleza aleatoria de los gestos en el conjunto de entrenamiento, no permitió diferenciar entre algunas figuras como óvalos y círculos, o cuadrados y rectángulos.

Al realizar las pruebas de campo se notó que los ejercicios presentan un reto mayor para los estudiantes cuya discapacidad cognitiva viene asociada con una discapacidad motriz, y que sin embargo, son estos pacientes los que mas motivados se muestran para realizar los ejercicios; esforzándose para realizar movimientos que les resultan difíciles de ejecutar. También se notó que para estudiantes con retraso mental leve y sin ninguna discapacidad motriz, es fácil realizar los ejercicios en los primeros niveles de dificultad, y aunque en algunos casos no implique un reto significativo para el paciente, siempre muestran un interés particular asociado con la forma de interactuar con la aplicación.

La implementación de la aplicación informática y las pruebas con pacientes mostraron que se pueden incorporar tecnologías interactivas en la educación especial, se observó que tanto pueden aportar en este entorno de aprendizaje, y el gran interés que genera en los pacientes una herramienta que les permita interactuar con la música e interpretarla, a la vez que provee una realimentación visual y sonora.

Según las observaciones realizadas por los especialistas de la Fundación Procer, los ejercicios trabajan principalmente la atención, la concentración y la coordinación. Están diseñados para alumnos con retraso mental entre moderado y grave, y que presenten discapacidades motrices moderadas. En este grupo de personas los ejercicios músico-terapéuticos además de motivar la realización de movimientos que les resultan difíciles, los impulsa a estar atentos para saber en qué momento deben realizar los movimientos indicados.

Se concluyó que el ejercicio de realizar figuras geométricas con las manos trabaja las capacidades motrices finas y se evidenció que es importante, para facilitar el aprendizaje del paciente, que el reconocimiento de las figuras geométricas sea

independiente del sentido de ejecución del movimiento; permitiendo que los pacientes que lo realicen no deban tener conocimientos sobre lateralidad.

La especialista en educación para personas con trastornos neuro-motores de la fundación Procer recomendó implementar ejercicios que involucren reconocimientos de poses del cuerpo completo para trabajar la psicomotricidad gruesa, y desarrollar ejercicios que obliguen al usuario a realizar movimientos rápidos con sus extremidades.

10. REFERENCIAS BIBLIOGRAFICAS

[1] Black, D., Gohlke, K. (2010). FoleySonic: Placing Sounds on a timeline through gestures. *Audio engineering society convention paper num 8004.*

[2] Rosemberg, Craig. (1993). Future Human interfaces for computer controlled soundsystems. *Audio engineering society convention paper num 3735.*

[3] Churnside, A., Pike, C., Leonard, M. (2011). Musical Movements Gesture Based Audio Interfaces. *Audio engineering society convention paper.*

[5] Bevilacqua, F., Dobrian, C. (2010). Mapping sound to human gestures: demos from video-based motion capture systems.

[6] Puckette, M. (2004). Programming electronic music in pure data, (1), 12-14.

[7] Magee, W. (2006). Electronic technologies in clinical musical therapy. *Technology and Disability*, (18), 139–146.

[8] Bevilacqua, F. (2006). Gesture control of music systems*, Ircam Real time Music interactions Team website.*

[9] Sze, S. (2005). Effects of music therapy in children with disabilities.

[10] Pelliteri, J. (2000). The consultant's corner: Music therapy in the special education setting. *Journal of Educational & Psychological Consultation, (14), 45-52.*

[11] Hunt, A., Ross, Kirk., Neighbour, M. (2008). Multiple Media Interfaces for Music Therapy*, IEE Multimedia,* (22), *50-58.*

[13] *Abhishek, K. (2011).* Skeletal Tracking using Microsoft Kinect.

[14] Chang, Y., Chen, S., Huang, J. *(2011).* A Kinect-based system for physical rehabilitation: a pilot study for young adults with motor disabilities.

[15] Wobbrock, J., Wilson, A., Yang, L. *(2007).* Gestures without libraries or toolkits training: a $1 Recognizer for User interface Prototypes.

[16] Reyes, M., Dominguez, G. *(2011).* Feauture Weighting in Dynamic Time Warping for gesture recognition in depth data.

[17] Brasher, H. *(2012).* Microsoft Kinect Gesture recognition software for stroke patients. *Inside technology*, (9), 2-6.

[18] Legos Project. IRCAM WebPage, *http://legos.ircam.fr/.*

[19] Bevilacqua, F., Zamborlin, B., Sypniewski, A., Schnell, N., Guédy, F., Rasamimanana, N. (2009). Continuous realtime gesture following and recognition, *Gesture in Embodied Communication and Human-Computer Interaction*.