

Visound, reporte intermedio post piloto

Ionatan Perez

2 de agosto de 2016

Índice

1. Introduccion	2
2. Del vOICe al Visound	2
3. Ideas preliminares y desarrollo de software	3
4. Experimento Piloto	6
4.1. Setup experimental	6
4.2. Procedimiento	10
4.3. Resultados	10
5. Proximo experimento	12
5.1. Discusion previa:	12
5.2. Propuesta general:	12
5.3. Setup experimental.	12
5.4. Resultados y mediciones esperadas.	13

1. Introduccion

Visound es el nombre que elegi para la aplicacion con la que queremos estudiar cuestiones de geometria en el contexto de la tecnologia vOICe.

La tecnologia vOICe transforma imagenes en sonido con el objeto de que personas ciegas puedan percibir estímulos visuales de su entorno mediante sustitucion sensorial. Para eso lo que hace es codificar la coordenada horizontal en tiempo (hace un barrido de izquierda a derecha de la imagen) y la vertical en frecuencia (los pixeles altos suenan agudo y los bajos suenan grave). Esta tecnologia permite transformar imagenes monocromaticas arbitrariamente complejas en sonido. En terminos fisicos interpreta cada columna de una imagen como la transformada de fourier del sonido a representar en un lapso de tiempo y luego concatena los sonidos correspondientes.

Este documento es una puesta al dia de:

- Cuestiones tecnicas relacionadas a la adaptacion de la tecnologia vOICe
- Hipotesis iniciales de trabajo
- Repaso de pruebas preliminares y diseño del experimento piloto
- Resultados del experimento piloto.
- Objetivo proxima medicion
- Diseño proximo experimento

2. Del vOICe al Visound

Para poder realizar los experimentos era necesario contar con un algoritmo que transformara las imagenes en sonidos, de manera de disponer de estímulos visuales y sus correspondientes representacion sonora. Primero intentamos emular (porque no teniamos acceso al original) la logica del visound, pero esto presento dos problemas importantes al aplicarlo a nuestros estímulos. El hecho de que los estímulos con los que queriamos trabajar fueran estímulos sencillos formados por segmentos rectos en combinación con la pixelacion inherente al algoritmo, resulto en saltos de fase (entre pixel y pixel) en señales que debian ser continuas y en batidos producto de que una misma linea que ocupara mas de un pixel de ancho podia generar dos frecuencias muy similares.

Decidimos entonces (considerando que solo vamos a usar figuras formadas por segmentos rectos) reemplazar el algoritmo del vOICe por uno ad-hoc que a partir de la informacion conceptual de los segmentos creara las rampas de sonido correspondientes y luego las combinara (al igual que en la imagen) los sonidos. Para esto era fundamental contar con algoritmos que generaran imagenes en forma conceptual.

El primer paso fue crear un codigo que a partir de parametros adecuados a cada diseño experimental creara archivos SVG con la informacion de los segmentos a dibujar. A partir de estos archivos es facil crear imagenes porque SVG es basicamente un XML estandarizado para el cual hay librerias en todos los lenguajes que lo interpretan y representan en formato grafico. Lo que hicimos nosotros fue crear un codigo en JAVA que por un lado leyera el SVG en busca de los segmentos rectos y por otro que a partir de esos segmentos generara las rampas de sonido correspondientes (y las combinara en un unico audio por imagen).

Un problema inmediato que surgio fue el de los segmentos verticales. Porque no pueden ser representados como una rampa. Primero probamos crear a partir de los segmentos verticales un pulso de sonido limitado en frecuencia (lo que conceptualmente representaria un segmento vertical) pero no fue una solucion satisfactoria porque al oido suena muy diferente que un segmento casi vertical (el pulso en frecuencia tiene colas en el tiempo que se notan cuando incluye frecuencias bajas), y eso introduce una diferencia cualitativa intrinseca a una orientacion arbitraria y especifica que hubiera interferido con los experimentos. Lo que decidimos entonces fue emular los segmentos verticales como segmentos 'cuasi verticales' es decir con mucha pendiente, tanta

que no se pudiera diferenciar si se trata de un segmento ascendente o descendente. En esa instancia todavia no disponiamos de experimentos adecuados para medirlo, pero el criterio utilizado fue que si el 'ancho' del segmento es menor a 0.01 (en unidades arbitraria que especificamos mas adelante) lo extendiera en dicho valor. La pendiente de una recta no solo depende del 'ancho' sino tambien del 'alto', pero en general el la longitud de los segmentos no es inferior a las 50 unidades el angulo minimo (respecto a la vertical) resulta menor a los $0.01/50$ radianes. En los resultados posteriores (figura COMPLEATAR) se puede ver que este angulo esta efectivamente por debajo del umbral de deteccion tipico para segmentos verticales.

Otra cuestion a mencionar es que en el codigo que transforma imagenes en sonidos se establecen los parametros fisicos que determinan las transformacion. Mantener una consistencia en estos parametros es muy importante porque afectan sustancialmente la transformacion de imagen a sonido y como estos se perciben, que son el objeto de estudio de nuestro trabajo. Estos parametros son:

- La frecuencia maxima (la que corresponde al borde superior), inicialmente fue establecida en 8000hz y luego bajada a 4000hz a partir de quejas de los usuarios (era realmente muy agudo) y un estudio de ingenieria inversa sobre la app de vOICe que determino que ellos usan de tope 5000hz
- La frecuencia inferior que fue situada en 100hz porque es un numero que esta rasonablemente por encima del minimo audible tipico de humanos. Al hacer la ingenieria inversa en la app vOICe no fue facil determinar el limite usado por ellos.
- Las dimensiones predeterminadas de la imagen (100x100 en unidades arbitrarias (puede que el SVG asuma que son pixeles))
- Si la escala es fija o no. Osea, si se asigna al 0 de altura la frecuencia 100Hz y al 100 la frecuencia 4000Hz mas alla de las dimensiones reales de la imagen o si se redimensiona la imagen para que un borde sea 100Hz y el otro 4000Hz. Dado que siempre usamos imagenes de 100x100 en nuestro caso es irrelevante la eleccion
- La duracion predeterminada de los audios, y la relacion pixel/segundo (osea la medida horizontal esperada de las imagenes). Se eligio que las imagenes de 100x100 duren 5 segundos. Y se respeto eso en todos los experimentos.
- Que se use una escala logaritmica en el mapeo de altura a frecuencia y la base de dicha escala (usamos 10).
- La frecuencia de sampleo del audio generado (usamos 44100Hz)

Algo que esta pendiente de revision y seria bueno ver antes de largar los proximos experimentos es normalizar de alguna manera el volumen relativo segun la frecuencia. En este momento todas las frecuencias tienen el mismo volumen, y la sensacion es que los agudos se escuchan mas fuerte que los graves. Lo cual no es bueno.

3. Ideas preliminares y desarrollo de software

La idea de este trabajo es estudiar la percepcion de conceptos geometricos en el marco de la sustitucion sensorial tipo vOICe. Originalmente la idea era ver hasta que punto se pueden percibir conceptos categoricos como paralelismo, formas (angulos rectos, agudos u obtusos), simetrias, etc. Como dependen estas detecciones de diferentes parametros (por ejemplo la orientacion de la figura) y hasta que punto se puede entrenar la capacidad y transferir el aprendizaje al variar parametros del estimulo.

Para eso hacia falta:

- Generar estímulos visuales con su correspondiente estímulos auditivos
- Como la idea es disponer de estímulos controlados donde se modifiquen algunos parametros especificos necesitabamos generarlos desde cero mediante algun algoritmo, para poder controlar cuantitativamente

las variaciones y para poder generar cientos de estímulos diferentes en forma rápida y eficiente. Para eso desarrollamos un código que generara la secuencia de parámetros necesarios para realizar cada experimento, otro código que a partir de los parámetros conceptuales de los segmentos (ubicación y orientación) generara los archivos SVG (archivos de texto en formato XML) que es un formato estandarizado de dibujo vectorial. También necesitamos un código que a partir de los SVG transformara esa información en un archivo de audio (ver sección 2).

- Generar una plataforma (o app) que permitiera realizar el experimento con usuarios.

La plataforma (elegimos LibGDX, un framework basado en JAVA pero que permite compilar a JAVA, a HTML5 y a Android) debía proveer toda la interacción entre el experimento en sí y el usuario. Se tenía que poder mostrar al usuario estímulos (visuales o auditivos) y luego dar la opción al usuario de registrar algún tipo de respuestas. También, por las características dinámicas de los experimentos planteados, la app debía poder adaptar los estímulos a las respuestas de los usuarios. Si bien parece una tarea sencilla, el proyecto de aplicación resultó ambicioso y el desarrollo completo de la app llevó cerca de 8 meses. El proceso no fue lineal, durante el mismo se fue adaptando, rediseñando y mejorando tanto a nivel funcional como conceptual la aplicación e implicó un gran proceso de aprendizaje personal en relación a la programación.

- Diseñar a nivel conceptual los experimentos.

Más allá de tener la aplicación, realizar el experimento requiere un trabajo de diseño, en los estímulos, en la dinámica de aplicación de los mismos, en los datos a medir y en el postprocesamiento de los datos. El diseño de los experimentos fue cambiando con el tiempo. Con versiones preliminares realizamos pruebas preliminares para comprobar que las hipótesis de trabajo fueran razonables y para ir adaptando el experimento a realizar en sujetos experimentales según parámetros y respuestas que observamos en pruebas preliminares (basicamente realizadas en mí)

El experimento inicial planteado fue ver la capacidad de distinguir entre las siguientes categorías:

1. Ángulos : Distinguir si un sonido corresponde a un ángulo agudo recto u obtuso
2. Paralelismo : Distinguir si un sonido corresponde a dos rectas paralelas o no
3. Cuadriláteros : Distinguir si un sonido corresponde a un cuadrado o un rombo

Para luego entrenar en alguna de ellas y evaluar la mejora en la performance de la tarea.

Las pruebas las realicé en mí mismo y fue claro que cierta capacidad de distinción existía pero que dependía mucho de las figuras en particular y sobre todo de la orientación. La verdad que no fui muy sistemático en el registro de resultados y reconstruir ahora los resultados me resultaría difícil porque fueron todas pruebas preliminares y no muy planificadas (error a tratar de corregir en el futuro!) Pero en su momento sirvió para: 1) Ver que más allá de las limitaciones efectivamente había capacidad de interpretar los estímulos y medir cosas! 2) No tenía mucho sentido usar como medida mis propias mediciones porque el efecto del entrenamiento en mí mismo era muy obvio 3) Era fundamental antes de realizar alguna medición buena de aprendizaje calibrar la dificultad de los estímulos.

El siguiente experimento que intenté hacer fue medir la dificultad en función de la orientación del estímulo. Esto implicó un cambio conceptual en el diseño de la aplicación (lo que llevó como un mes de mejoras y adaptaciones) porque para hacer el experimento anterior se podía tener una serie de estímulos prearmados y mostrarlos en algún orden random. Luego simplemente había que hacer un estadístico para saber si las respuestas eran significativamente diferentes que una respuesta random (lo cual no fue trivial porque cada trial o estímulo aceptaba diferente cantidad de respuestas posibles). El nuevo experimento requería medir un umbral de detección. La idea era ver cuán diferente tenía que ser un ángulo de un ángulo recto para ser diferenciado y lo mismo con rectas paralelas.

Para eso habia que diseñar un algoritmo dinamico tipo quest. El problema ademas es que al estar programada la aplicacion en JAVA no podia implementar ningun algoritmo ya armado en matlab y cualquier cosa que quisiera hacer la tenia que reproducir yo.

Lo que hice fue armar un algoritmo ad-hoc que basicamente hacia lo siguiente:

Categorizar los estímulos segun la intensidad de la señal a medir (en el caso de los ángulos fue la diferencia entre el ángulo formado y el recto, en el caso del paralelismo fue el ángulo formado por las rectas) y luego ordenarlos y numerarlos segun esa categoria. A esa numeracion lo denomine nivel de señal. La idea de numerar y no usar la señal original fue que pudiera no ser lineal la escala de medicion con respecto al parametro fisico que la genera. Esto es porque la cantidad de estímulos crece muy rapidamente con la cantidad de variables y definicion en las mismas, y por cada recurso utilizado crece el tamaño de la app (los recursos no se generan en tiempo real), y utilizando una excala con mayor densidad de estímulos en la zona que probablemente estuviera el umbral se maximizaba la capacidad dinamica de deteccion. En el caso de paralelismo se utilizo una escala logaritmica y en el de ángulos un paso discreto grande lejos de los ángulos criticos (0,90 y 180) y otro mas chico cerca de eso ángulos.

Presentar al usuario los estímulos de manera intercalada (ángulos agudos y obtusos o rectas que se juntan en una u otra direccion) comenzando con lo de mas alta señal (muy no rectos o muy no paralelos), y si el usuario detecta la señal disminuir su intensidad, en caso que no aumentarla. De esta manera la idea es que mientras el usuario distingue el experimento se vuelve cada vez mas complicado hasta que no distingue mas.

Originalmente pensamos en intercalar señales con estímulo con señales sin estímulo (recto o paralelas) pero el problema de eso es que es muy facil generar memoria y que el reconocimiento no fuera del aspecto a medir en el estímulo sino del estímulo como un todo. Por eso necesitamos tener en cada caso dos alternativas similares e intercalables (agudos y obtusos o convergentes hacia un lado o convergentes hacia el otro) para que el usuario este obligado en cada trial a detectar la señal en forma activa. De esta manera por cada experimento obtuvimos dos curvas de aproximacion.

Algunas otras cuestiones tecnicas que se consideraron en el diseño del algoritmo fue que el paso inicial fuese decreciendo con el tiempo, para acelerar la convergencia al nivel de umbral (y que no este el usuario un monton de trials en la zona donde detecta bien). Lo que se considero fue una disminucion en el paso cada vez que el usuario respondia un trial mal. No fue la mejor eleccion, probablemente convenga hacer que el salto dependa del numero de trials previos, porque a veces resulto que un error al inicio de la secuencia hizo que la convergencia se enlenteciera sin sentido. Por otro lado para evitar un 'random walk' se puso como criterio para disminuir la señal que los dos ultimos trials fueran correctos, y solo uno correcto para aumentar la señal. Esto fuerza al algoritmo a salir de la zona de no deteccion en caso de que haya un ingreso casual y establece como limite estadistico el nivel de señal donde hay dos tercios de chance de reconocer el estímulo.

Otra cosa que se considero pero que convendria cambiar es que el algoritmo se corte en el momento en que detecta una convergencia, para evitar trials innecesarios y maximizar el rendimiento del tiempo (un problema inicial fue que el experimento era muy largo). Esto hizo que algunas secuencias se corten muy rapido y la verdad es que no resulto muy claro que establecer como criterio de convergencia. En el futuro la idea es hacer todas las secuencias igual de largas.

Con este diseño en mente y una version en desarrollo hicimos un experimento en el cual quisimos ver la dependencia del umbral de deteccion de paralelismo con respecto a la inclinacion de las rectas. En el siguiente link estan los detalles. El experimento resulto extremadamente largo (como 4hs) por lo que solo lo hice yo dos veces y conseguí que lo haga una vez mi hermana. El resultado se puede ver en las figuras 1 y 2

Un experimento similar (pero con la fuerte hipotesis del resultado) fue realizado para los ángulos. En este caso se realizo el experimento para 19 orientaciones diferentes (ver figura 3) donde se agrupo las referencias similares para evitar acostumbamiento y cada referencia tiene 4 curvas de aproximacion, una por ángulos agudos y otra por obtusos al ángulo recto de 90 grados y al de 270 grados. Este experimento fue realizado solo para verificar la posibilidad factica de hacerlo y porque el software y diseño experimental necesario era basicamente igual al que se planeaba usar en el experimento piloto. La prueba la realice en mi una vez y los resultados se pueden observar en los graficos 5, y 4.

Separacion angular minima (en $^{\circ}$) que se puede detectar en funcion de la orientacion de las rectas 'pseudoparalelas'.
El color del punto representa si se considera que la medicion paso un test de confianza o no.

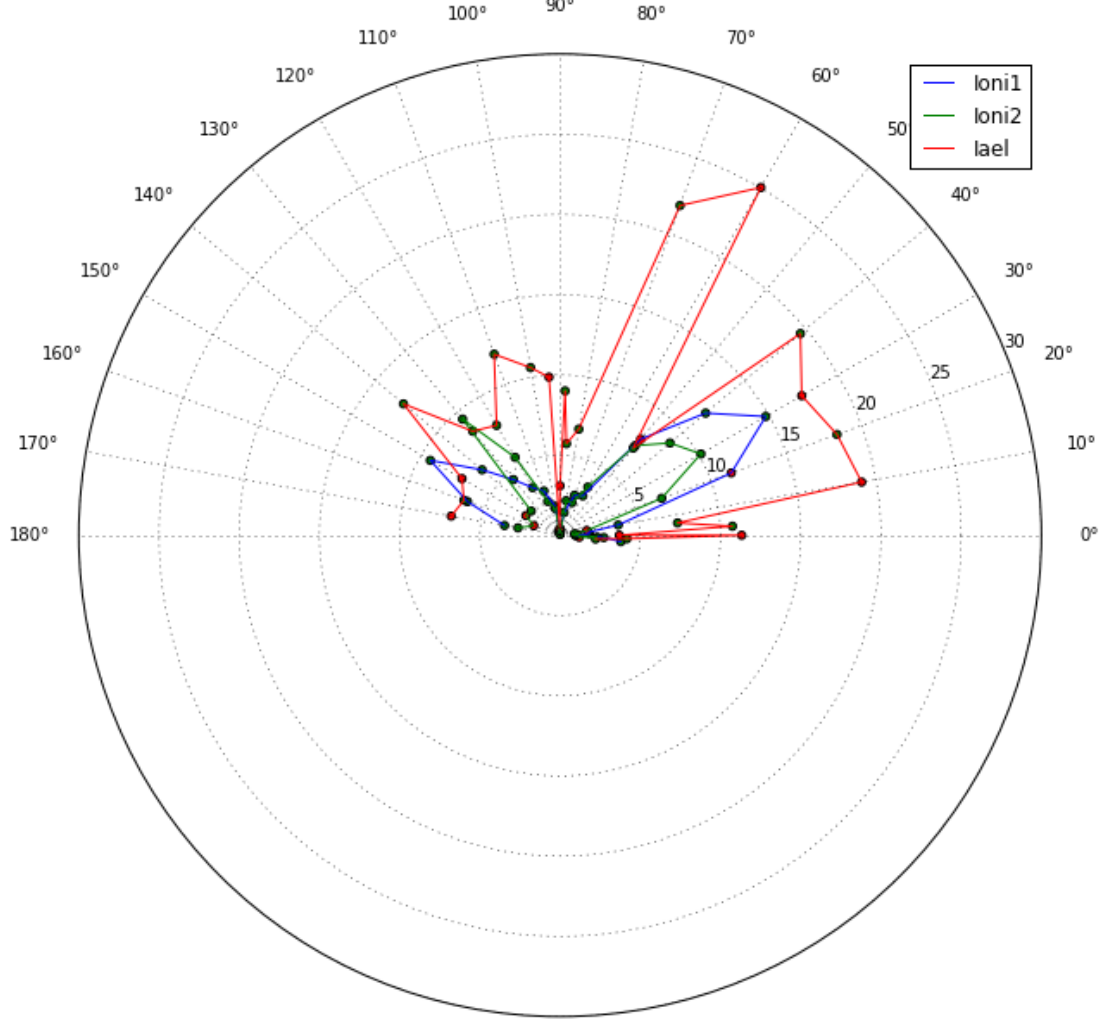


Figura 1: Umbral de deteccion de paralelismo/no paralelismo en funcion de la orientacion de las rectas.

4. Experimento Piloto

Con la idea de tener una calibracion de dificultad para una posterior medicion de aprendizaje (no tiene sentido medir aprendizaje si no sabemos si lo que hay que aprender es trivial, imposible o algo accesible de aprender) y tambien con la idea de poner a punto la dinamica y logistica de medir con sujetos experimentales decidimos hacer un experimento piloto que midiera el umbral de deteccion de paralelismo y angulos para ciertas orientaciones que luego se pudieran usar en el experimento siguiente. Otra ventaja de realiza este experimento es que podiamos validar con datos experimentales no sesgados (por hacer los experimentos en mi mismo) los resultados obtenidos hasta el momento.

4.1. Setup experimental

El codigo se puede descargar de este link

El experimento conto con 9 niveles. Tres iniciales de tutorial, tres de evaluacion de paralelismo y tres de

Separacion angular minima (en $^{\circ}$) que se puede detectar en funcion de la orientacion de las rectas 'pseudoparalelas'.
El color del punto representa si se considera que la medicion paso un test de confianza o no.

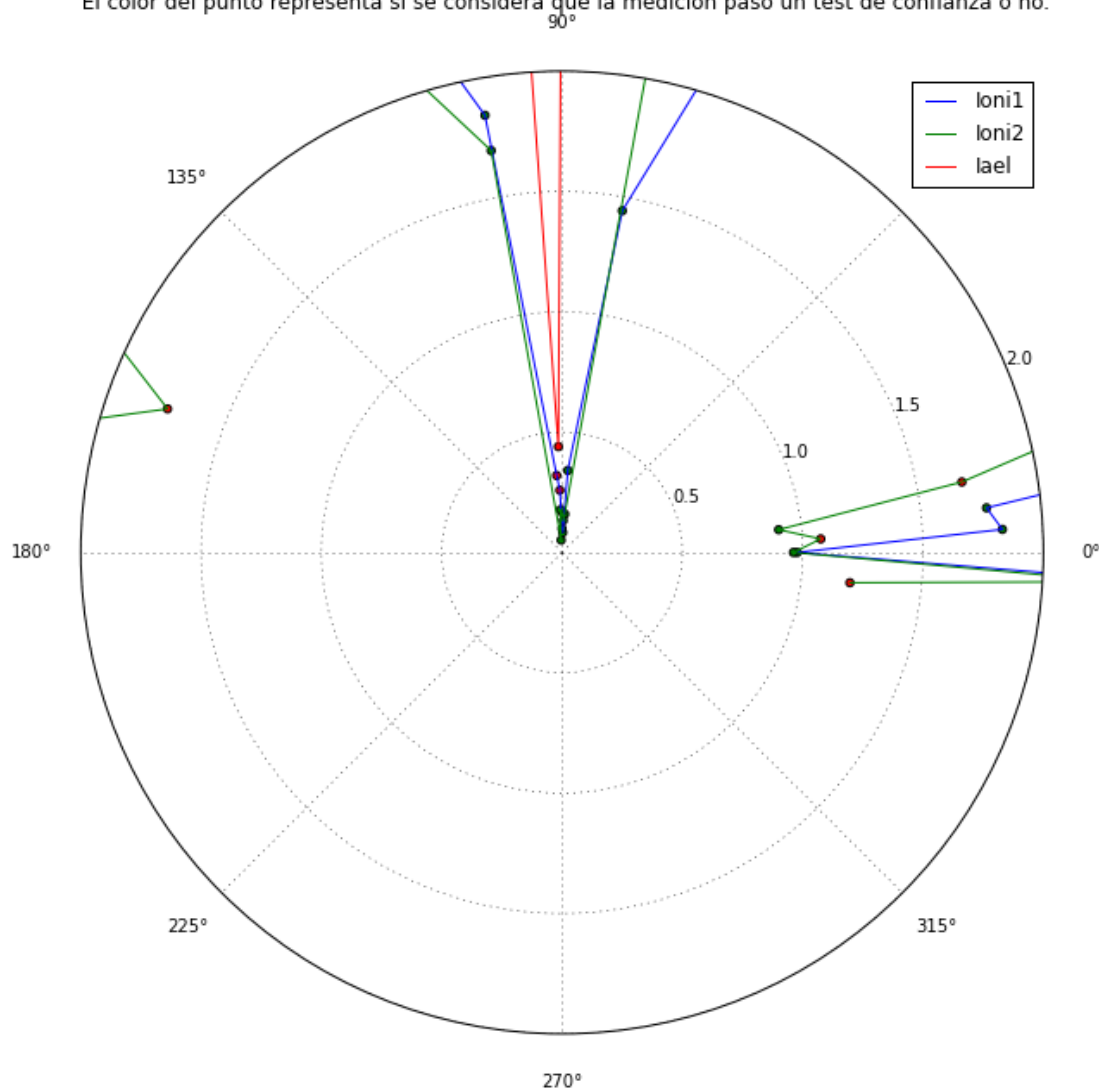


Figura 2: Zoom del umbral de deteccion de paralelismo/no paralelismo en funcion de la orientacion de las rectas, para observar valores en orientaciones vertical y horizontal.

angulos.

Los tutoriales, que deben ser completados previamente a los niveles de verdad, consistian, el primero en cuatro trials con imagenes de segmentos que se pueden tocar para escuchar como suenan. El segundo y el tercer en niveles de ejemplo de los que se realizaran luego, pero con feedback. La idea fue que el usuario primero pueda reconocer como suenan imagenes y luego poner a prueba brevemente su criterio sin estar tomando mediciones. En el tutorial se utilizo angulos de referencia diferente a los posteriores. De los tutoriales no se extrajo ninguna medicion, si bien el ejemplo de paralelismo y de angulos envia los datos como si fuese un nivel mas (pero que no fue procesado posteriormente)

Para los niveles que si se busco medir se eligio el siguiente criterio:

- Se uso tres niveles diferentes con orientacion, 0° , 30° , 90° . Porque queriamos contrastar la sensibilidad

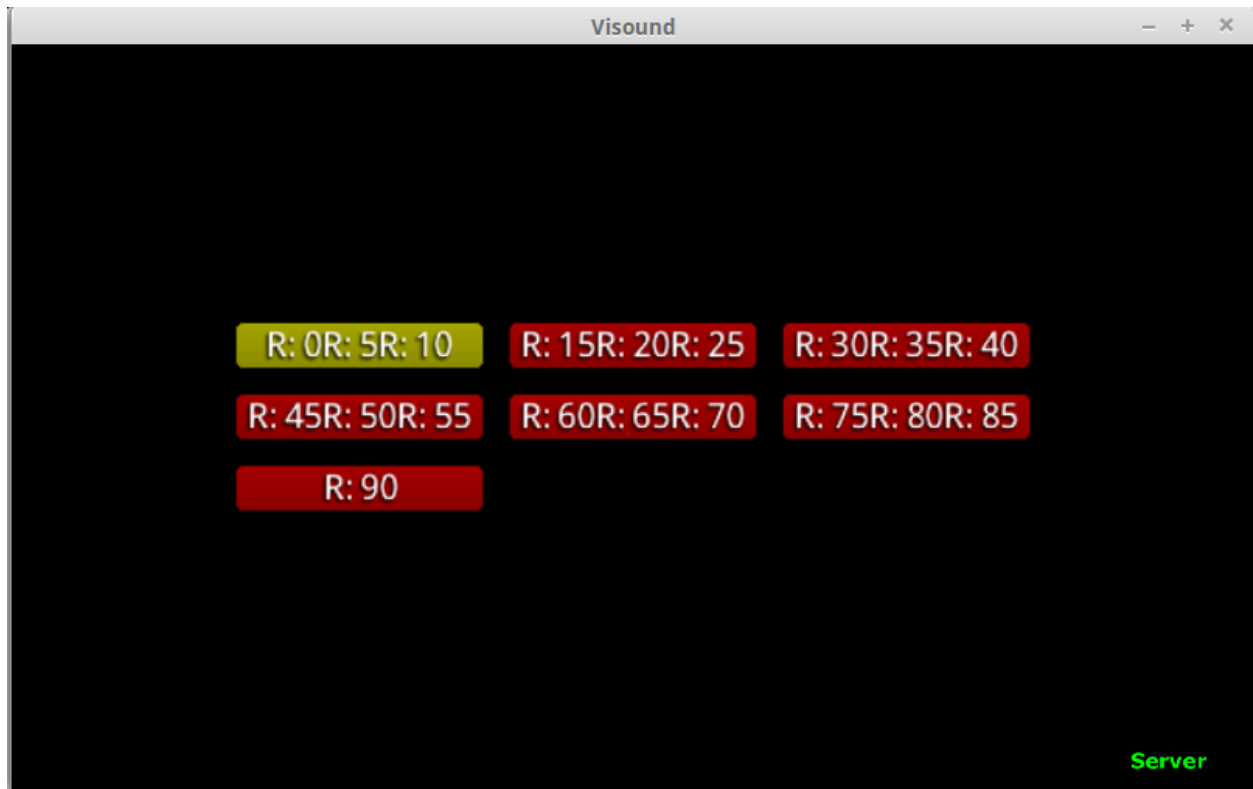


Figura 3: Imagen de la app en su version para hacer un analisis preliminar de la sensibilidad a la deteccion de angulos en funcion de la orientacion de referencia. Se puede observar 19 niveles agrupados por referencias similares. Cada referencia posee 4 curvas de aproximacion.

en los ejes que esperabamos fuera mucho mayor que cualquier otra, pero tampoco iguales entre si, pues conceptualmente la informacion relevante esta representada en una caracteristica ditinta del estimulo (tiempo o frecuencias).

- En cada nivel se midieron dos curvas de aproximacion, en el caso de los angulos desde los agudos y obtusos y en el de paralelismo convergiendo hacia uno u otro lado. Cada una de esas curvas podia tener una longitud variable, con un maximo de 40 trials por nivel, pero si una curva convergia antes de llegar a los 40 trials en el nivel se la consideraba concluida. Para evaluar la convergencia se tomo en cuenta el promedio del nivel de señal (que es un indice, no el valor fisico) y si la desviacion estandar en las ultimas mediciones (se tomo 6) es menor que un cierto valor (se tomo 0.5)
- En el caso de los angulos se genero estímulos con saltos de 10 grados de variacion del lado movil excepto cerca de los ejes donde se creo angulos con lados a 1, 2,4,6 y 8 grados de los ejes. Esto fue para poder evaluar con cuidado en la zona donde esperabamos mayor sensibilidad en la deteccion. Probablemente un error haya sido que al elegir el lado fijo en 30°, el angulo recto y las dos señales mas cercanas difieren en 10°. En el caso de los experimentos de paralelismo se creo una escala logaritmica de angulos crecientes que fueron de 0.1 a 25 grados (hacia cada costado, en total el angulo formado es el doble) con una escala de 40 pasos.
- Para reconstruir una curva de aproximacion, se hizo que dos o mas respuestas correctas consecutivas disminuyan el nivel de señal, pero una incorrecta lo aumente. Ademas cada vez que luego de una correctas se respondia una incorrecta el salto entre dificultad y dificultad (originalmente en 5) se hacia menor hasta llegar a la minima unidad.

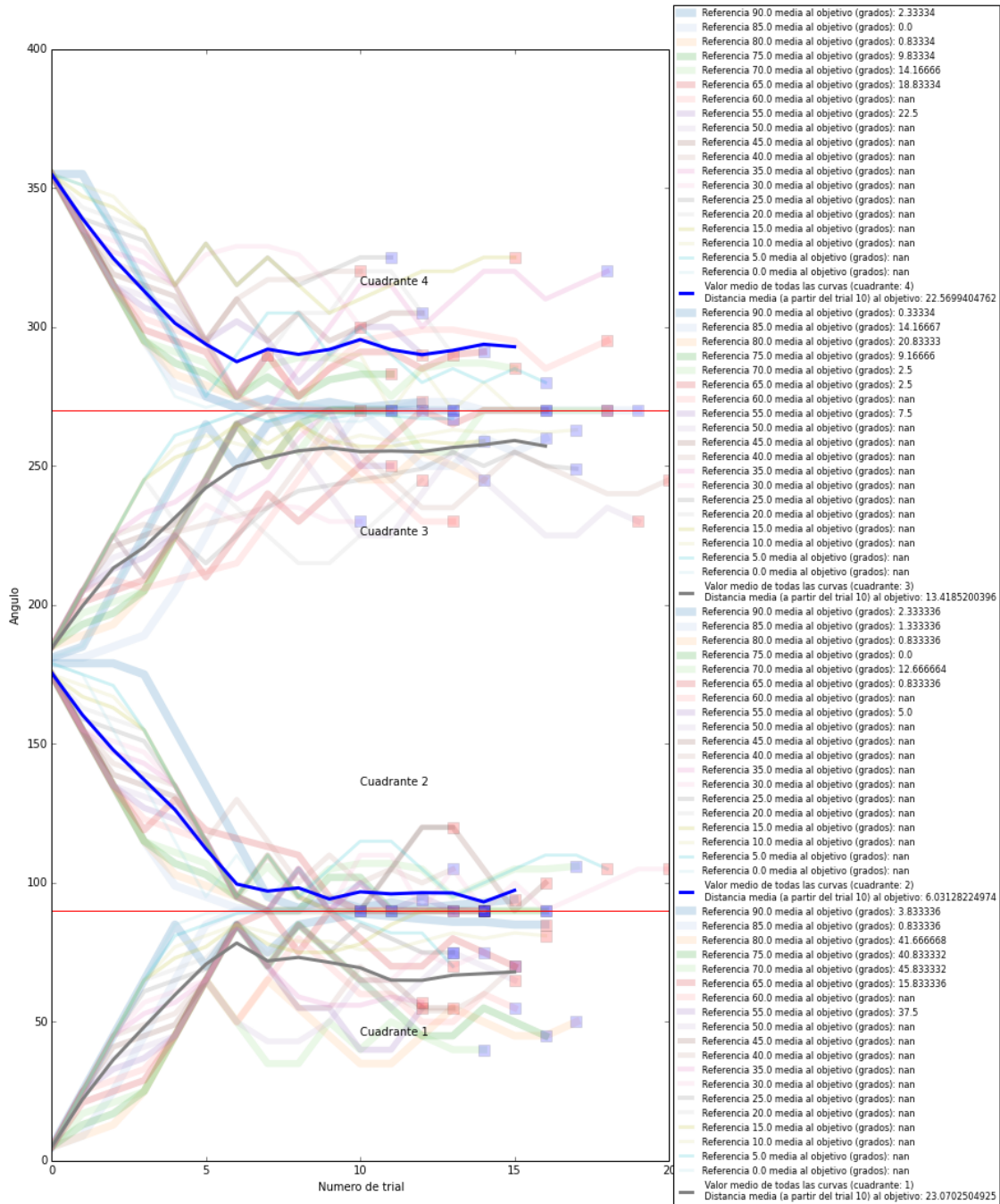


Figura 4: Curvas de aproximacion del la señal de estimulo en funcion del trial realizado. Si el usuario responde mal a la categoria (agudo/obtuso) la señal (o angulo formado) se aleja del angulo recto, si responde bien se acerca. Se puede observar todas las curvas de aproximacion donde cada color respresenta una orientacion diferentes para el lado fijo. sobre las lineas individuales so observa un promedio de todas las referencias. El hecho de que el cuadrante 2 sea el que mayor sensibilidad posee lo atribuimos a que es el unico donde nunca se superponen los sonidos correspondientes a ambos lados del angulo.

- De cada serie se registro una lista de datos correspondientes a cada trial donde figura: nivel de señal, nivel de estimulo, si se respondio bien o no, datos de la version del software, datos del nivel y experimento, señal relativa a la referencia. Probablemente un error haya sido no guardar ademas de si se acerto o no, cual fue la respuesta marcada porque en el caso de los angulos hay mas de dos opciones posibles.

4.2. Procedimiento

El experimento se realizo con 14 sujetos. Se convoco postulantes a traves de una base de datos de mails de alumnos interesados de UTDT y hablando con conocidos que estaban dispuestos a participar. Los alumnos de UTDT no tenian conocimiento previo del experimento mas alla de una breve presentacion en el formulario de inscripcion, los conocidos en algunos casos si tenian un leve conocimiento del contexto y el objetivo, pero no se lo considero un problema, ya que el protocolo incluia una introduccion que contara los objetiso y el contexto en el momento de realizar el experimento.

Los sujetos vinieron al Laboratorio de Neurociencia de UTDT donde realizamos el experimento en una laptop (prestada por matemarote) con auriculares, dentro de la sala de experimentos. Los primeros sujetos usar auriculares ear-in los ultimos unos con parlante acolchado. Como presentacion les contabamos el marco en el cual se realiza el proyecto, como funciona la logica del vOICE y algunos ejemplos con palabras, como iba a ser el experimento (haciamos una mencion explicita a que es lo que se queria medir porque por el diseño del experimento es muy explicito y no tenia sentido ocultarlo), que tenian que hacer el tutorial, y luego acompañabamos a los sujetos hasta que hubieran realizado el tutorial (al principio nos quedamos los tres niveles, con los ultimos sujetos nos retiramos apenas se sintieran comodoss con el software).

Una vez finalizada la sesion (que duro entre 30 y 50 minutos segun el sujeto) les pedimos que completaran esta encuesta cuyos resultados estan disponibles aqui

4.3. Resultados

Los resultados del procesamiento de los datos pueden descargarse de este link donde figura el codigo en python con el que procesamos los datos guradados, las copias de la base de datos descargadas y los graficos obtenidos.

Para analizar los datos se intento responder las siguientes preguntas

1. ¿El usuario entendio el sistema de estímulos y es capaz de detectar las señales, o responde al azar?
2. En caso de que detecte, ¿interpreto bien los estímulos o los interpreto sistematicamente mal?
3. En caso de que los dos puntos anteriores se den, hasta donde lleo a detectar estímulo el sujeto.

Para responder la primer y segunda pregunta lo que necesitamos ver es si hay diferencia significativa entre las respuestas que da el sujeto, y un sujeto que responde al azar. Hicimos un test de significancia contra una probabilidad binomial en el caso del paralelismo o trinomial en el caso de los angulos, y observamos cuanto vale el p-valor correspondiente a cada una de las series de datos (2 por cada nivel jugado). En caso de que el p-valor de inferior al 0.05 significa que el usuario responde como si detectara correctamente el estímulo muy por encima de lo que indicaria la estadística. En este caso asumimos que los datos son validos. Vale destacar que es de esperar que al llegar al umbral las respuestas se vuelvan azarosas (con un funcionamiento correcto del experimento) por lo que un umbral de $p=0.05$ en realidad es un criterio mas robusto de lo usual. Si por el contrario el p-valor es muy alto significa que el usuario responde sistematicamente mal. Esto lo asociamos con el hecho de que el sujeto haya detectado los estímulos, pero los interprete incorrectamente asignandolos a la categoría contraria a la que corresponden. En este caso consideramos que era un limite razonable usar un p-valor de 0.9 como limite, es decir que haya una probabilidad menor al 0.1 de que se trate de un resultado azaroso. Por ultimo las series de datos que tienen un p-valor entre 0.05 y 0.9 las consideramos casos donde el sujeto no logro detectar sistematicamente la señal. Como la configuracion experimental es tal que frente

a una respuesta azarosa la señal de estímulo es grande, asumimos que estos sujetos o no entendieron la consigna, o no están pudiendo identificar en los sonidos las características necesarias.

Los resultados de este análisis pueden observarse en la figura 6 donde están agrupados los análisis según el experimento realizado. La secuencia en que aparecen los resultados coinciden con la secuencia en que los sujetos realizaron los niveles. Al respecto cabe destacar que un usuario hizo notar que el tercer nivel (paralelismo vertical) es un muy fuerte entrenador de la detección de segmentos verticales que ayuda a mejorar la performance en la detección de los niveles 4 y 6 (o sea ángulos donde uno de los dos lados es vertical o cercano a la vertical). De los resultados hay dos datos remarcables. Por un lado que los niveles en los que se observa un alto nivel de detecciones incorrectas son el primero y el último. En el primero se podría relacionar a que el tutorial no sea lo suficientemente largo y entonces haya individuos que tardan en acostumbrarse a caracterizar correctamente lo que escuchan, pero lo más relevante es la tasa de detecciones falsas en el último nivel. En este nivel uno de los dos lados del ángulo (el que no varía) está vertical, en la mitad superior de la imagen. Algo que habíamos notado en las pruebas cualitativas (y de hecho fue ratificado reiteradamente por los usuarios al charlar en el primer trial del tutorial) es que dos segmentos verticales que solo difieren en su altura son muy difíciles de diferenciar. Es difícil ubicar el conjunto de frecuencias que suenan simultáneamente dentro de un espectro con referencias absolutas. En particular, en el nivel 6 (ángulos a 90 grados) el segmento fijo ocupa la mitad superior de la pantalla, pero podría confundirse con uno que ocupe la mitad inferior, y en ese caso los estímulos corresponderían a la categoría contraria. Al diseñar el experimento consideramos este problema pero decidimos dejarlo así para probar precisamente si se notaba en los resultados, y efectivamente los resultados parecen respaldar la hipótesis de trabajo.

El segundo punto a destacar es que en la orientación 30 grados de paralelismo hay un muy marcado índice de señales donde los sujetos no pudieron identificar correctamente los estímulos. Esto no es sorprendente porque es el experimento donde más marcado aparece el problema de que no hay elementos cualitativos para distinguir las respuestas con un criterio que no sea cuantificar el nivel de inclinación de los estímulos. En todos los demás experimentos distinguiendo alguna característica de los estímulos (por ejemplo si un lado tiene pendiente positiva o negativa, etc) se puede pensar conceptualmente la imagen y responder, mientras que en este experimento en particular hay que distinguir exclusivamente cual de los dos segmentos tiene más pendiente, sin recurrir a interpretaciones o lógica alguna. Si bien el resultado fue muy marcado, hay que mencionar que el algoritmo de convergencia, en conjunción con el criterio de detección hace que este valor pueda estar sobredimensionando. Algunos casos donde haya habido detección para señales muy altas, pero no para señales más bajas, podría no reflejarse en los resultados como un caso de detección debido a que, si se ve que se converge muy rápido al umbral hay una gran proporción de respuestas azarosas 'correctas' y esperadas.

Una vez descartados los datos donde no consideramos que haya habido detección correcta procedimos a cuantificar el nivel de umbral. Esto se puede observar en las figuras 7 y 8 donde unificamos todos los valores de umbral de los diferentes usuarios correspondientes a un mismo nivel para establecer un nivel de umbral en función de la orientación. Como es cualitativamente diferente la aproximación por una u otra serie (según de qué lado se acerque al valor neutro la señal) buscamos el nivel de umbral aproximando la señal desde ambos 'lados'. Y también en conjunto. Un problema que surgió al hacer este análisis, es que en algunos casos el umbral no estaba representado por una distribución normal o donde los valores estuvieran agrupados en forma que se pudiera obtener un valor medio representativo. Paso en varios de los experimentos que la moda era muy marcada, pero había valores muy lejanos. Interpretamos que esto está relacionado con que ciertos sujetos usaron un mejor criterio que otros y por lo tanto hubo una respuesta cualitativamente diferente en algunos casos. Para analizar esto lo que hicimos fue por un lado evaluar los resultados descartando los outsiders y por otro incluyéndolos. Se observa que no siempre coinciden los resultados.

5. Proximo experimento

5.1. Discusion previa:

La idea original era usar los resultados del experimento piloto para tener una calibración de dificultad a partir de la cual armar un experimento de aprendizaje que utilizara figuras complejas y/o en direcciones arbitrarias haciendo un estudio de detección de categorías. Pero luego de la charla decidimos reenfocar el experimento en el aprendizaje sin mezclar el tema de la detección de categorías. Utilizar los mismos experimentos del piloto pero para estudiar aprendizaje y transferencia, y de paso agregar una cuantificación de confianza.

5.2. Propuesta general:

La idea sería realizar un experimento donde midamos el umbral en el cual el usuario no puede distinguir paralelismo de no paralelismo, ni un ángulo del recto, para 4 orientaciones diferentes, y observar si ese umbral decrece con el entrenamiento y si una vez mejorada la habilidad, se transfiere entre orientaciones. También queremos registrar niveles de confianza.

5.3. Setup experimental.

Convocaríamos sujetos (en principio 8) a una serie de sesiones (en principio 6) pagas de una hora aproximadamente cada una. Cada sesión consistiría en lo siguiente:

1. Tutorial y test de umbral inicial con un setup muy similar al del piloto, usando 8 niveles con orientaciones 30, 60, 120 y 150 grados en ángulos y paralelismo.
2. Entrenamiento con/sin feedback en una de las orientaciones. ¿Se podría hacer que en la mitad de los casos coincida la orientación de ambas categorías mientras que en la otra mitad no? Quizás esto haga que la estadística tenga un N demasiado bajo. Pero sino no podemos independizar entrenamiento entre categorías. Quizás convenga hacer 16 sujetos con un entrenamiento de media hora, en lugar de una y que la mitad entrenen ángulos y la otra mitad rectas.
3. Continuar entrenamiento. Con/Sin feedback
4. Continuar entrenamiento. Con/Sin feedback
5. Continuar entrenamiento. Con/Sin feedback
6. Test igual al inicial (sin el tutorial) en todas las orientaciones.

En el test cada nivel contaría con 30 trials de aproximación desde cada lado. Donde luego de marcar la respuesta elegida se debe indicar la confianza en la respuesta elegida sobre una escala continua. Todo sin feedback.

En los niveles de entrenamiento el procedimiento sería similar pero con un número mucho mayor de trials (podrían ser 100 de cada lado), y hay que ver si tiene sentido consultar confianza en todos los trials o solo en un subconjunto. También me gustaría revisar el criterio con que se da feedback. Una ventaja de dar feedback una de cada dos sesiones es que se puede medir como evoluciona el umbral en función de si se da feedback o no, pero una contra es que si el usuario está muy perdido puede desperdiciar una sesión entera. Una alternativa intermedia puede ser alternar secuencias de 25 trials con feedback con otros 25 sin.

Un cambio que habría que hacer con respecto al setup del experimento piloto es que en la categoría ángulos ya no tiene mucho sentido preguntar por ángulos rectos. Esto se justificaba cuando había estímulos sobre los ejes donde la sensibilidad era muy alta, pero la verdad que terminé trayendo algunos problemas (basicamente que si el usuario estaba convencido que llegó al recto y es cierto nunca sale de ese estímulo). Por otro lado también distorsiona un poco la estadística, porque en un caso hay que responder entre tres opciones

y en otro entre dos. Además eran diferentes las escalas de los estímulos, mientras que en el paralelismo había una escala de dificultad logarítmica, para el de caso de ángulos era de paso constante, lo cual no tiene mucho sentido. La propuesta es reducir el número de opciones a ángulo agudo y obtuso y hacer una lista de estímulos en escala logarítmica entre un ángulo grande y uno muy pequeño.

Otro cambio que haría que hacer (en el código más que nada) es que registre junto a la respuesta dada, cuánto se tardó en responder desde que empezó el estímulo y cuánto en dar un índice de confianza. Además hay que implementar el botón para medir confianza.

5.4. Resultados y mediciones esperadas.

A diferencia del experimento piloto donde se buscaba medir un nivel de umbral general para todos los usuarios acá la idea es hacer una comparación inter usuario con respecto a la situación inicial. Esto permitiría cuantificar la evolución del aprendizaje en relación a la cantidad de estímulos recibidos y a si los mismos son con feedback o sin él, y también correlacionar con el nivel de confianza en las respuestas dadas. Esto para cuantificar el aprendizaje en orientación en que se entrene el usuario. Por otro lado, comparando los umbrales del test inicial y final (donde el usuario realiza un test de umbral en todas las orientaciones y categorías) se puede cuantificar el nivel de transferencia que hay en el aprendizaje entre la orientación entrenada y las demás, observando cuán efectiva es la transferencia en situaciones de proximidad, de simetría y de variación de categoría.

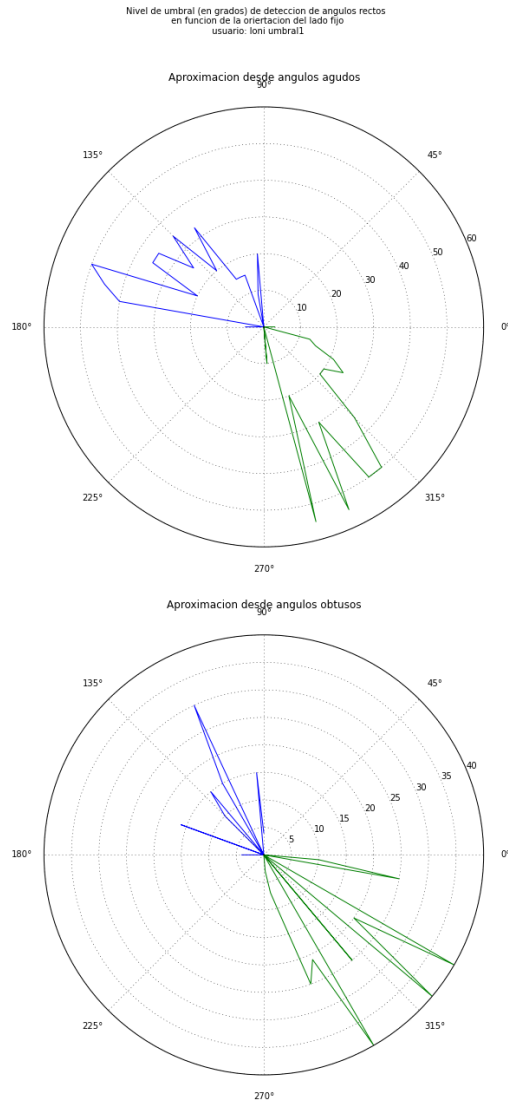


Figura 5: Sensibilidad en función de la orientación. Las curvas representan el ángulo 'umbral' marcando la dirección del lado móvil. Un gráfico representa la curva cuando el ángulo se aproxima al recto desde los agudos y la otra desde los obtusos. Esperábamos que se observara un marcado efecto de pérdida de sensibilidad cuando el lado fijo no coincidiera con un segmento vertical u horizontal, sin embargo se ve que hay medidas donde esto no sucede. La explicación más factible es que al ser un experimento preliminar realizado en mí mismo, donde conocía los estímulos y además, debido a un error de diseño, el paso era tal que respondiendo bien los primeros estímulos la señal llegaba justo al ángulo recto, es probable que pudiera estimar cuando se trataba del ángulo recto no por detectarlo sino por contexto. Sin embargo llama la atención la asimetría entre la aproximación aguda y obtusa al respecto. Se observa que en los ejes en ningún caso la sensibilidad resultó baja.

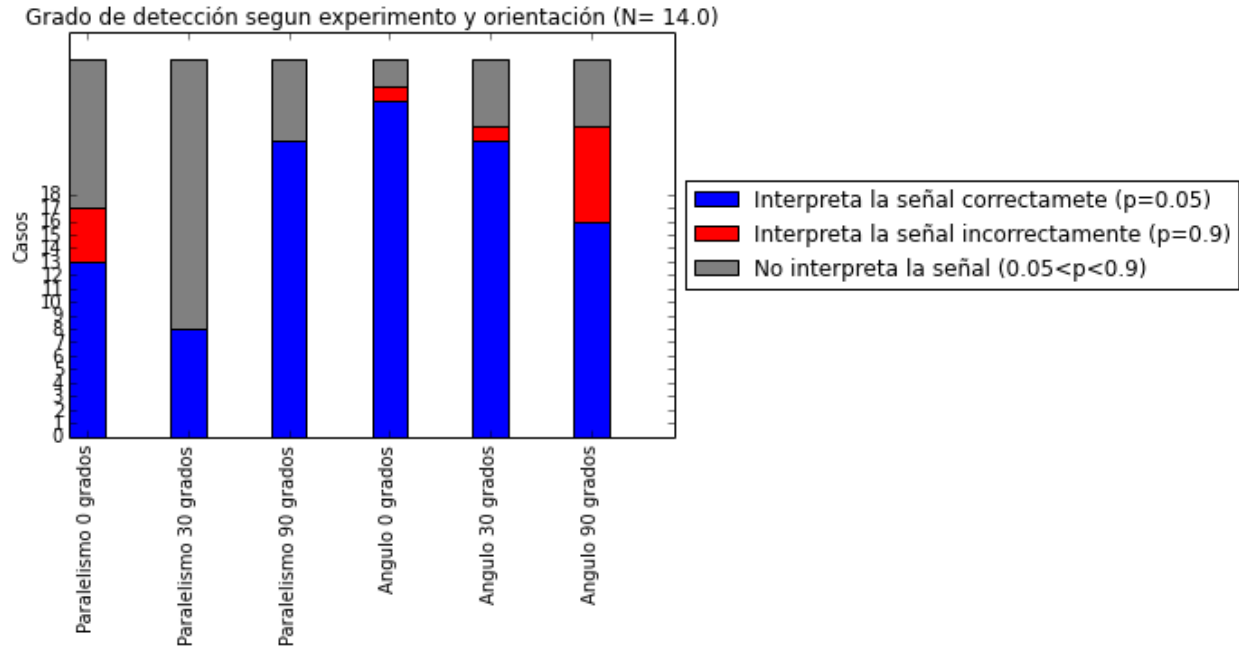


Figura 6: Analisis de significancia para cada curva de aproximacion. Cada nivel incluye dos curvas. Si el valor de p es mayor a 0.9 se considera que el sujeto reconocio los estímulos pero los interpreto incorrectamente. Si es menor a 0.05 que lo hizo correctamente. Se destaca el alto valor de series que no llegaron a la significancia en paralelismo a con orientacion de 30 grados y la alta tasa de detecciones incorrectas para angulos a 90 grados.

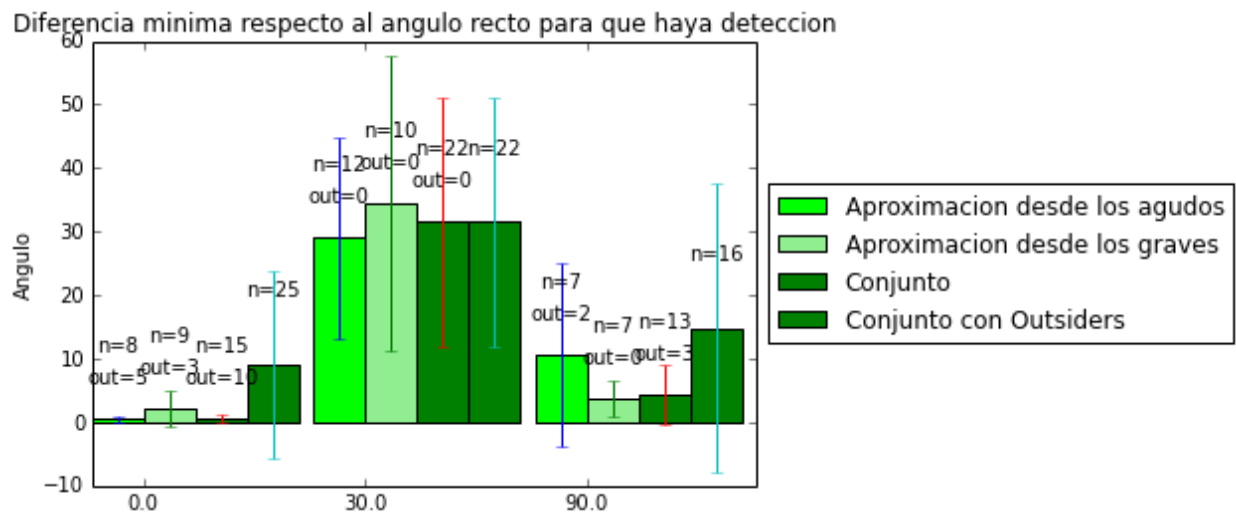


Figura 7: Umbral de deteccion donde se pide distinguir un angulo recto de uno agudo u obtuso según diferentes orientaciones. La señal medida es la diferencia respecto al angulo recto que debe haber para que los estímulos se vuelvan indistinguibles. Se calcula incluyendo y excluyendo datos outsiders. El hecho de que el umbral para la orientacion 30 grados sea 30 grados indica que los sujetos no pudieron distinguir mas alla de reconocer cualitativamente de que lado de la vertical queda el lado movil.

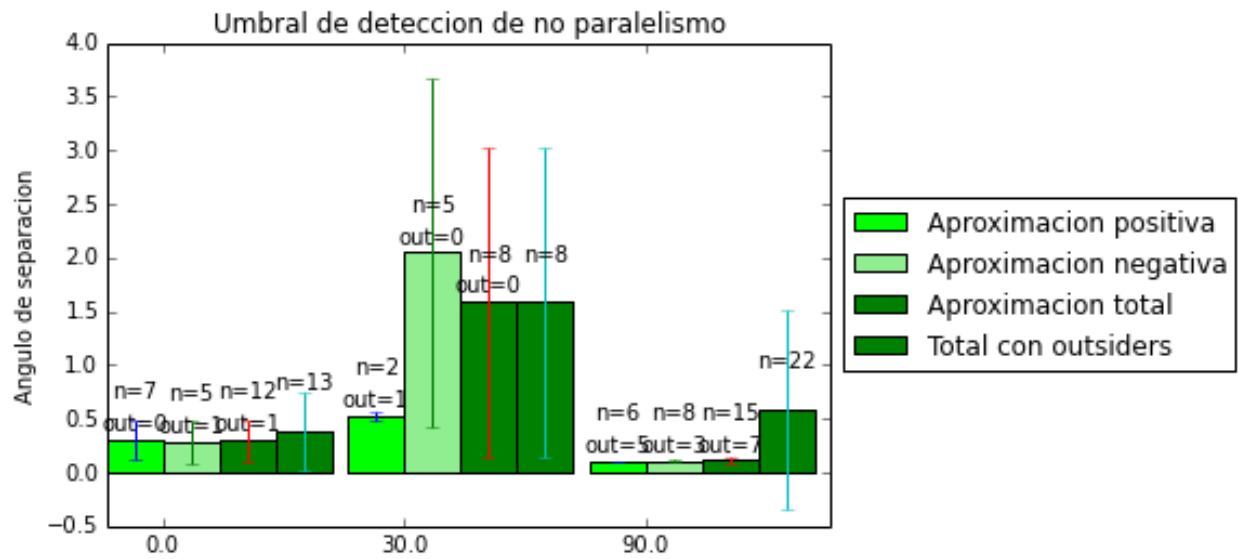


Figura 8: Umbral de deteccion (en funcion de la orientacion) donde se pide distinguir en que sentido convergen dos rectas cuando se presenta estímulos cada vez mas paralelos. La señal medida es el angulo entre las rectas y la bisectriz del angulo formado por ambas rectas (que no se llegan a juntar en los estímulos). Se observa una marcada diferencia entre las orientaciones asociadas a los ejes (donde los sonidos de cada segmento son cualitativamente distintos) y la orientacion de 30 grados donde se debe percibir una característica exclusivamente cuantitativa y la sensibilidad fluctua mucho de individuo a individuo. También se observa que en la orientacion de 90 grados hay un conjunto de individuos con mucha sensibilidad y otros que estan muy lejos (notar la dispersion, dado que el valor medio esta muy influenciado por los valores bajos).