GENERACIÓN DE UNA HERRAMIENTA QUE PERMITA ESTIMAR EL FOCO DE ATENCIÓN VISUAL DE CADA UNA DE LAS PERSONAS QUE SE ENCUENTRAN EN UN SALÓN DE CLASES POR MEDIO DE TÉCNICAS DE VISIÓN ARTIFICIAL

RAFAEL ERNESTO POVEDA LASPRILLA

UNIVERSIDAD PEDAGÓGICA NACIONAL
DEPARTAMENTO DE TECNOLOGÍA
LICENCIATURA EN ELECTRÓNICA
BOGOTÁ
2013

GENERACIÓN DE UNA HERRAMIENTA QUE PERMITA ESTIMAR EL FOCO DE ATENCIÓN VISUAL DE CADA UNA DE LAS PERSONAS QUE SE ENCUENTRAN EN UN SALÓN DE CLASES POR MEDIO DE TÉCNICAS DE VISIÓN ARTIFICIAL

RAFAEL ERNESTO POVEDA LASPRILLA

Trabajo de grado para optar al título de Licenciado en electrónica

Directora
ANGÉLICA VELOZA SUAN
Ingeniera de Sistemas

UNIVERSIDAD PEDAGÓGICA NACIONAL
DEPARTAMENTO DE TECNOLOGÍA
LICENCIATURA EN ELECTRÓNICA
BOGOTÁ
2013

Nota de aceptación

_____

_____

_____

_____

_____

Firma

Nombre:

Presidente del jurado

_____

Firma

Nombre:

Jurado

_____

Firma

Nombre:

Jurado

Bogotá, Mayo 28 de 2013

CONTENIDO

Pág.

1. INTRODUCCIÓN 8

1.1. Identificación del problema 8

1.2. Justificación 9

1.3. Delimitación 9

1.4. Objetivos 11

1.4.1. Objetivo general 11

1.4.2. Objetivos específicos 11

2. ANTECEDENTES 12

3. MARCO DE REFERENCIA 14

3.1. Interfaz natural de usuario 14

3.2. Kinect 14

3.2.1. Cámara RGB 15

3.2.2. Cámara de profundidad 16

3.2.3. Micrófonos 17

3.3. Kinect SDK 20

3.3.1. Rastreo de esqueleto 20

4. DESARROLLO DEL PROYECTO 22

4.1. Realización de pruebas 22

4.2. Selección de la cámara 23

4.2.1. Limitaciones del Kinect 25

4.3. Captura de imágenes 26

4.4. Encontrar personas en la escena 29

4.5. Encontrar la cabeza de las personas 34

4.6. Determinar el ángulo del cuerpo 35

4.7. Determinar el ángulo de las cabezas 37

4.8. Exportar imágenes 38

4.9. Procesamiento con Matlab 38

5. Pruebas 46

5.1. Desempeño 46

5.2. Trabajo futuro 47

5.3. Deficiencias 48

6. Conclusiones 49

BIBLIOGRAFÍA 50

LISTA DE FIGURAS

Pág.

1 Grados de libertad de la cabeza 10

2 Resolución del angulo de guiñada a reconocer 11

3 Fotografía del kinect 15

4 Cámaras del kinect 16

5 Micrófonos del kinect 17

6 Ángulos de visión del kinect 18

7 Técnica de luz estructurada 18

8 Patrón de puntos infrarojos del kinect sobre una escena 19

9 Resumen del algoritmo de Body Tracking 21

10 Etapas de desarrollo del proyecto 22

11 Posibles cámaras a usar en el proyecto 24

12 Imagen de profundidad dañada por distancia 25

13 Representación de un pixel de profundidad 26

14 Gráfica de profundidad contra valor de gris en la imagen de salida 29

15 Imagen de profundidad del kinect 30

16 Sustracción de fondo 31

17 Histograma de una imagen de la cabeza de una persona a 2 metros y 3 metros 32

18 Histograma de una imagen normalizada de la cabeza de una persona a 2 metros y 3 metros 34

19 Sustracción de fondo con imagen normalizada 35

20 Puntos identificados del cuerpo por el Kinect SDK 36

21 Imágenes de cabezas encontradas en una imagen de profundidad 36

22 Imagen de la cabeza de la misma persona a 1.50m y a 2.27m 37

23 Distancia de los hombros con respecto al sensor 38

24 Representación de la distancia entre hombros 39

25 Planteamiento de ecuaciones para hallar el angulo del cuerpo 40

26 Herramienta hallando en tiempo real el ángulo del cuerpo 40

27 Herramienta hallando en tiempo real el ángulo del cuerpo para dos personas 41

28 Verdadero positivo 315 grados 42

29 Verdadero positivo 180 grados 43

30 Verdadero positivo 0 grados 44

31 Falso positivo 0 grados 45

1. INTRODUCCIÓN

Nosotros lo humanos hacemos uso de nuestros ojos y cerebro con el fin de percibir el mundo que nos rodea y tomar decisiones con base en los estímulos que recibimos. La visión artificial es la disciplina que trata de emular el comportamiento de la visión humana para que una máquina pueda utilizar esta habilidad con el fin de llevar a cabo tareas como la identificación de patrones y la extracción de características.

Aunque se ha avanzado mucho en el campo de la visión artificial , los sistemas existentes están lejos de ser perfectos o incluso de tener la misma precisión de la visión humana. Es por esto, que es importante que se realicen investigaciones que aporten al desarrollo de algoritmos y herramientas que permitan avanzar cada día mas en el perfeccionamiento del área.

Con el reciente lanzamiento de cámaras de bajo costo que permiten medir la distancia de los objetos en una escena [1], se ha abierto un nuevo sendero en el camino de la visión artificial el cual esta lleno de nuevos retos y posibles soluciones a problemas existentes.

La finalidad de este proyecto consiste en hacer uso de estas nuevas tecnologías emergentes en el estudio de la estimación del foco de atención visual de múltiples personas en una escena.

1.1. IDENTIFICACIÓN DEL PROBLEMA

La estimación del foco visual de las personas es un área de la visión artificial que ha tenido bastante auge en la última década por sus aplicaciones en las relaciones robot-humano. Para nosotros los humanos el determinar si una persona se comunica con

nosotros es una tarea que realizamos de forma inconsciente ya que nos entrenamos desde pequeños para saber cuando una persona se está dirigiendo hacia nosotros y cuando no, deduciéndolo de la posición de sus ojos y cabeza.

Los sistemas actuales que determinan el foco de atención visual se enfrentan a múltiples problemas entre los cuales destacan las imágenes en las que los detalles pequeños no son fácilmente identificables, el encontrar la cabeza de una persona en una imagen digital, y las condiciones variantes de iluminación. Todo estos inconvenientes demuestran la necesidad de encontrar alternativas, tanto en herramientas como en algoritmos, que permitan solucionar estos problemas de una mejor manera.

1.2. JUSTIFICACIÓN

La generación de una herramienta robusta que permita estimar el foco de atención visual en múltiples personas puede ser de utilidad para muchas aplicaciones en las que la interacción de humanos con máquinas no puede ser llevada a cabo de una manera tradicional (mouse y teclado) ya sea por limitaciones físicas de movimiento, como las que tienen personas con discapacidad, o por condiciones del entorno en las que no sea posible la interacción directa con las máquinas [2].

El desarrollo de una herramienta que permita estimar el foco de atención visual sirve como punto de partida para otro tipo de aplicaciones que hagan uso de interfaces alternativas al ratón y el teclado o el estudio del comportamiento humano en entornos no controlados.

1.3. DELIMITACIÓN

Investigaciones han demostrado que la determinación del foco de atención visual viene de una combinación de la posición de la cabeza y la dirección de enfoque de los ojos.

Por si misma, la cabeza puede dar una buena aproximación del foco de atención visual en situaciones en las que la información que proveen los ojos no está disponible, como por ejemplo, en imágenes de baja resolución o situaciones en las que la persona tiene gafas oscuras [3]. Por estas razones, el foco de atención visual de una persona va a ser considerado como la estimación de la posición de la cabeza.

Para los fines del presente trabajo, la cabeza será considerada como un cuerpo rígido desprendido del cuerpo, por lo que su movimiento se puede modelar con 3 grados de libertad: **cabeceo, alabeo, y guiñada**. Estos grados de libertad son los mismos que se usan para identificar la orientación de los aviones.

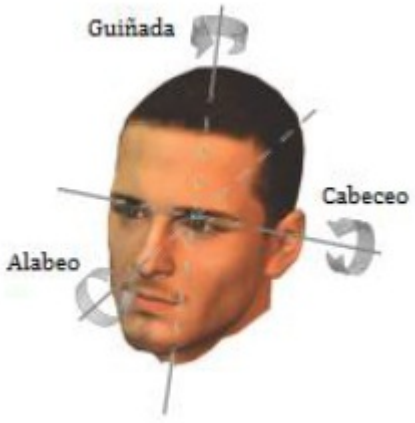

Figura 1: Grados de libertad de la cabeza

La herramienta reconocerá solamente los ángulos de guiñada con una resolución de 45 grados como se ve en la figura 2.

El lugar elegido para las pruebas fue un salón de clases con el fin de demostrar el desempeño del sistema a condiciones invariantes de luz y diferentes posiciones de las personas (de pie o sentadas).

Por último, y por limitaciones de hardware y software, las pruebas solo serán realizadas con máximo dos personas al mismo tiempo.

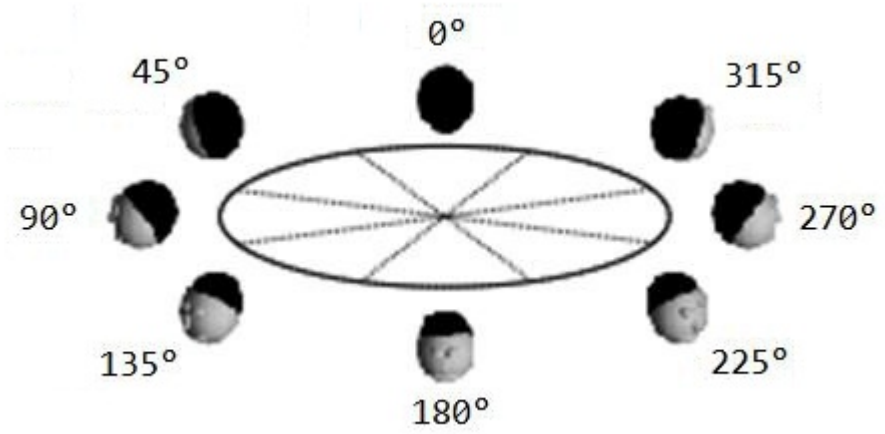

Figura 2: Resolución del angulo de guiñada a reconocer

1.4. OBJETIVOS

1.4.1. Objetivo general

Generar una herramienta que permita estimar el foco de atención visual de cada una de las personas que se encuentran en un salón de clases por medio de técnicas de visión artificial.

1.4.2. Objetivos específicos

1. Segmentar en una imágen, las cabezas de las personas que se encuentran en el salón de clases haciendo uso de una cámara de profundidad.
2. Determinar el ángulo de guiñada de cada cabeza en intervalos de 45 grados.
3. Elaborar una tabla que permita ver el ángulo de guiñada de cada cabeza por cada fotograma del vídeo.
4. Elaborar una interfaz gráfica de usuario que permita visualizar las imágenes de profundidad, la imagen RGB, y la tabla de los ángulos en función de los fotogramas.

2. ANTECEDENTES

Para llevar a cabo el proyecto, es necesario estudiar los trabajos relacionados que se han realizado en el pasado con el fin de sentar las bases del desarrollo que esta planteado. Aunque existen muchos artículos, investigaciones y trabajos de grado que podrían servir como punto de partida para el presente proyecto (muchos de los cuales se encuentran listado y clasificados en **??**), hay unos que se destacan por sus grandes similitudes frente al proyecto propuesto. Entre los trabajos mas cercanos destacan los siguiente:

Colour Invariant Head Pose Classification in Low Resolution Video

"Clasificación de la posición de la cabeza sin variaciones en el color en video de baja resolución" [4] es un artículo publicado por Ben Benfold y Ian Reid. Brinda excelente información sobre cómo obtener información de la posición de la cabeza en imágenes de 10x10 píxeles previamente seleccionadas y recortadas ubicando las cabezas para posteriormente usar técnicas de visión artificial para clasificar las imágenes entre categorías preestablecidas.

Real-Time Head Pose Estimation in Low-Resolution Football Footage

El trabajo *"estimación en tiempo real de la posición de la cabeza en videos de fútbol en baja resolución"*[5] realizado por Andreas Launila lleva a cabo una estimación de la posición de la cabeza de los jugadores en un campo de fútbol teniendo en cuenta factores ya conocidos como la posición del balón. Este trabajo especifica muy detalladamente los métodos usados para segmentar y posteriormente clasificar la posición de la cabeza de múltiples personas haciendo uso de maquinas de soporte vectorial.

Head Pose Estimation Using Stereo Vision For Human-Robot Interaction

"Estimación de la posición de la cabeza usando visión estéreo para interacción humano-

robot"[6] es un trabajo realizado por Edgar Seemann, Kai Nickel, Rainer Stiefelhagen en el año 2004. En él, presentan un método para estimar la posición de la cabeza por medio de una cámara estéreo. Este trabajo puede brindar aportes significativos al trabajo propuesto como el uso de imágenes con información de profundidad que permite contrarrestar los efectos negativos de la iluminación en un ambiente no controlado haciendo uso de redes neuronales para la clasificación de las imágenes.

Head Pose Classification in Crowded Scenes

"Clasificación de la posición de la cabeza en escenas con muchas personas"[7], es un trabajo realizado por Javier Orozco, Shaogang Gong, Tao Xiang. Presentado en la British Machine Vision Conference llevada a cabo en Londres en septiembre del 2009. Se enfoca principalmente en la detección de personas y su foco de atención visual en lugares concurridos comparando las imágenes obtenidas con imágenes plantillas previamente clasificadas.

3. MARCO DE REFERENCIA

3.1. INTERFAZ NATURAL DE USUARIO

La *interfaz natural de usuario* o NUI (Natural User Interface), es aquella interfaz en la que un usuario usa comportamientos naturales como hablar, mover su cuerpo, hacer gestos, etc. con el fin de interactuar con un sistema. Las interfaces naturales de usuario son una alternativa a la *interfaces gráficas de usuario* las cuales requieren para su interacción dispositivos como el mouse y el teclado.

Aunque las interfaces naturales de usuario parecen actuales gracias a la popularidad de los sistemas multitouch introducidos por el iPhone de Apple[8], sus primeras implementaciones se remontan al año 1979. En los laboratorios de investigación del MIT, el doctor Richard Bolt trabajaba en su proyecto llamado "Put-that-there: voice and gesture at the graphics interface" el cual consistía en el uso de comando de voz y gestos para controlar una interfaz gráfica[9]. Para que el proyecto funcionara era necesaria una habitación de aproximadamente 5 metros de ancho por 3 metros de largo y unos 2.5 metros de alto.

3.2. KINECT

El kinect es un accesorio para la consola de videojuegos de Microsoft, *Xbox 360*. Con su lanzamiento el 4 de noviembre del 2010 en Estados Unidos, se pretendía introducir a las interfaces naturales de usuario en el mundo de los videojuegos, lo cual ya había sido intentado de cierta manera por el *Wii remote* de Nintendo o el *Play Station move* de Sony.

Figura 3: Fotografía del Kinect. Imagen extraída desde:
http://www.vidaextra.com/juegos/accesorios/kinect

Lo que Microsoft no anticipó, fueron los diferentes usos alternativos que dieron los desarrolladores a este dispositivo trayendo consigo proyectos como *kinecthacks.net* o *openkinect.com* que tenían como finalidad mostrar proyectos realizados con el kinect que estaban lejos del uso pensado originalmente por su empresa creadora.

Los componentes principales del kinect son: Una cámara RGB, una cámara infraroja, un proyector infrarojo (Véase la figura 4), 4 microfonos (Véase la figura 5), ademas de un motor en la base que permite modificar el angulo de visión vertical.

3.2.1. Cámara RGB

La cámara RGB o cámara de color es la encargada de capturar y transmitir la información de color de la escena en frente del kinect. La información de color que transmite el kinect al computador es una sucesión de imágenes estáticas cuya tasa de transferencia depende de la resolución de la imagen. Por ejemplo, la cámara de color del kinect tiene una velocidad de 30 FPS (*frames per second* o cuadros por segundo) cuando la resolución de la imagen capturada es de 640 píxeles x 480 píxeles, mientras que la velocidad es de 12 FPS cuando la resolución de la imagen es de 1280x960. Es claro que mientras se aumenta la resolución de la imagen se disminuye la tasa de transferencia.

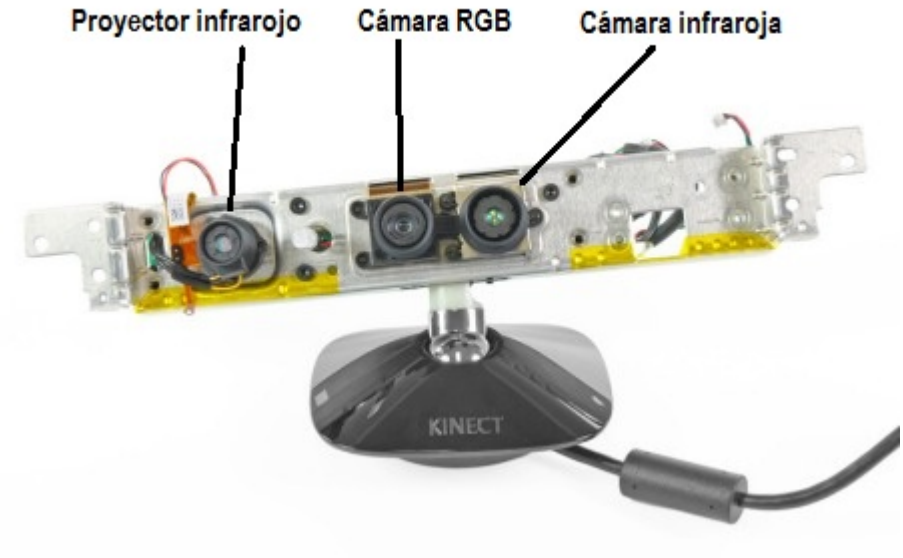

Figura 4: Disposición de las cámaras al interior del kinect. Imagen extraída desde: http://www.product-reviews.net/2010/11/05/kinect-teardown-pleo-the-dinosaur-robot-similarities/

El rango de visión de la cámara de color del kinect es de 43 grados verticales y de 57 grados horizontales como se aprecia en la figura 6.

3.2.2. Cámara de profundidad

La técnica usada por el kinect para conocer la distancia a la cual se encuentran los objetos en una escena es como conocida como *Luz estructurada*. Esta técnica consiste en proyectar un patrón conocido sobre una escena para luego ser capturado a través de una cámara infraroja y medir la deformación de este patrón (Figura 7). El kinect determina la distancia de los objetos midiendo las variaciones entre un patrón de puntos pseudo aleatorios que conoce y el patrón de puntos que captura (Figura 8). Una de las grandes ventajas de usar luz infraroja para medir las distancias de los objetos con respecto al kinect es la inmunidad que posee este sistema a las condiciones variantes de iluminación.

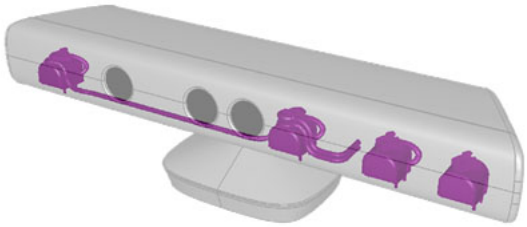

Figura 5: Disposición del arreglo de micrófonos al interior del kinect. Imagen extraída desde: http://goo.gl/SXwwE

De la misma manera que la cámara de color, la cámara de profundidad del kinect tienen diferentes resoluciones disponibles pero en contraste, todas las resoluciones tienen una tasa de transferencia de 30FPS por lo que la ventaja en utilizar imágenes con menor resolución radica en la velocidad de procesamiento. Las resoluciones disponibles para las imágenes de profundidad son 640x480 píxeles, 320x240 píxeles, y 80x60 píxeles. El rango de visión es el mismo que el de la imagen de color (43 grados verticales y de 57 grados horizontales).

3.2.3. Micrófonos

Adentro del kinect se encuentran ubicados 4 micrófonos cuya disposición es mostrada en la figura 5. El kinect utiliza estos 4 micrófonos para determinar el origen de una señal de audio como la voz de una persona. Para saber en que parte de la habitación se encuentra una persona hablando, cada micrófono escucha el sonido en cuestión y determina en que parte se encuentra midiendo el tiempo en que se demoro en llegar esa señal. Se puede inferir que en entre mas cerca este una persona de un micrófono en particular, este va a recibir el sonido mucha mas rápido que los otros micrófonos.

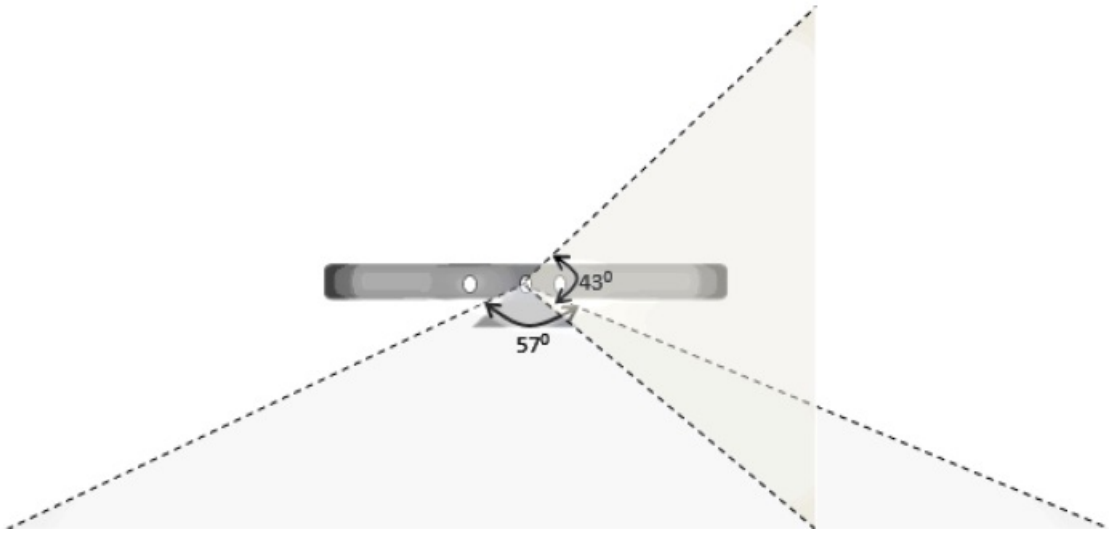

Figura 6: Angulo de visión vertical y horizontal de la cámara de color del kinect. Imagen extraída desde [10]

Figura 7: Ejemplo de uso de luz estructurada. Imagen extraída desde http://goo.gl/SYc0A

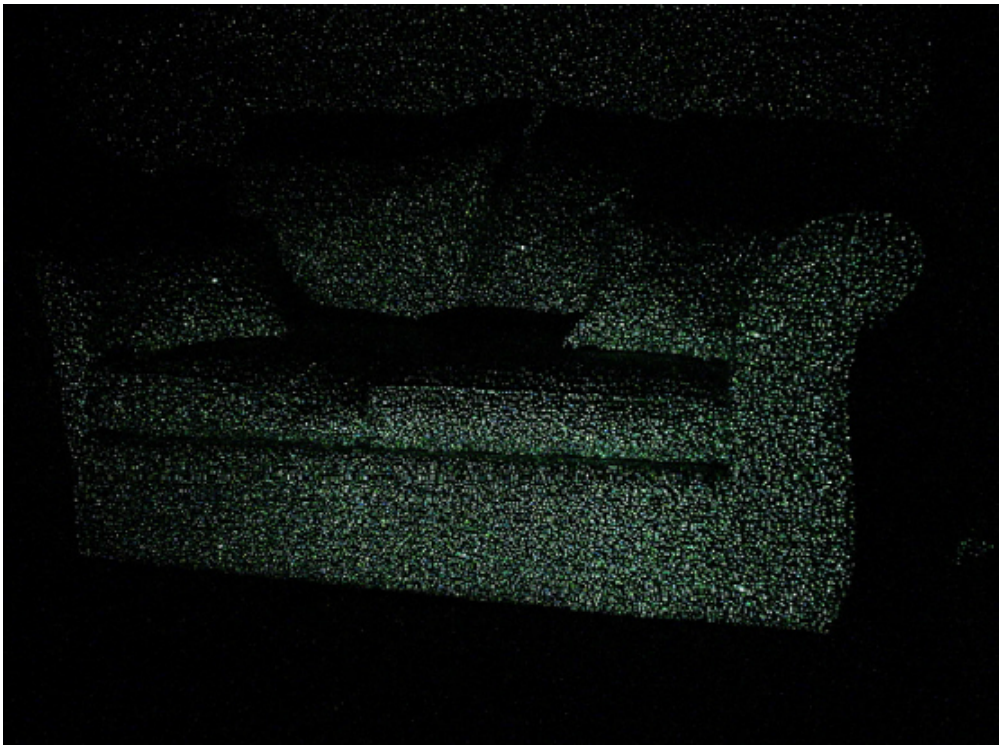

Figura 8: Imagen de un sofá iluminado por los puntos infrarojos emitidos por el Kinect. Imagen extraída desde [11]

3.3. KINECT SDK

El kinect SDK (Software Development Kit) es una colección de funciones que permiten el desarrollo de aplicaciones haciendo uso del kinect. Para poder crear aplicaciones que usen el kinect como dispositivo intermedio entre una interfaz natural y un usuario, es necesario contar con las siguientes herramientas:

- Microsoft Visual Studio 2010 (Las aplicaciones para el kinect pueden ser desarrolladas en Visual Basic, C++ y C#. Este ultimo fue elegido como lenguaje central para el desarrollo del proyecto).
- Microsoft .NET Framework 4.0 o superior
- Kinect for Windows SDK.
- Windows 7 o superior.

El kinect SDK provee las funciones necesarias para obtener las imágenes de color, las de profundidad, acceso a los microfonos del kinect y el rastreo de esqueleto que es la característica estrella de Kinect SDK.

3.3.1. Rastreo de esqueleto

El kinect SDK hace uso de la imagen de profundidad entregada por el kinect para llevar a cabo etiquetado de las partes del cuerpo de la persona que se encuentra en frente del kinect. La información entregada por el rastreo de esqueleto incluye las coordenadas en X, Y y Z de hasta 20 puntos del cuerpo humano. Con la información que se obtiene del Kinect SDK es posible saber la posición en la que se encuentran las distintas partes del cuerpo en un sistema de coordenadas relativas al Kinect. Los detalles del algoritmo usado por el departamento de investigación y desarrollo de Microsoft se pueden encontrar en http://research.microsoft.com/pubs/145347/bodypartrecognition.pdf.

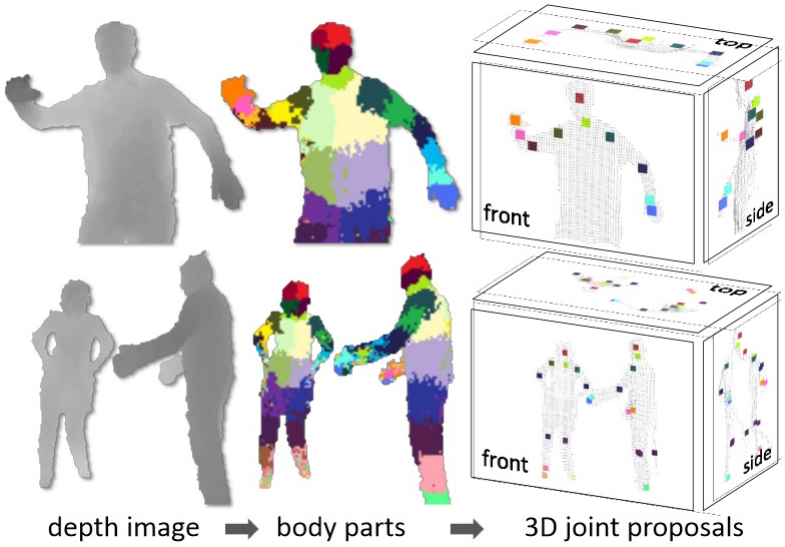

Figura 9: Resumen del algoritmo de Body Tracking

4. DESARROLLO DEL PROYECTO

Para el desarrollo del proyecto, se propone una metodología la cuál consta de seis etapas. Estas etapas son secuenciales y cada tiene como prerrequisito la anterior, de manera que, por ejemplo, no podrá realizar la etapa dos sin haber terminado la etapa 1. La metodología es presentada en la figura 10. Una vez acabadas todas las fases del proyecto se procederá a sacar las conclusiones sobre la herramienta y su evaluación en base al cumplimento de los objetivos propuestos.

Figura 10: Etapas de desarrollo del proyecto

4.1. REALIZACIÓN DE PRUEBAS

Aunque en la propuesta del proyecto se plantea la implementación de la herramienta en un ambiente no controlado como un salón de clases, se tomó la decisión de realizar una simulación del espacio propuesto por inconvenientes con la disponibilidad de los espacios. Por otro lado, y en pruebas realizadas preliminarmente, se pudo observar que la presencia de objetos como mesas y pupitres, funcionan en algunos casos como

interferencias al momento de realizar los pasos propuestos en la metodología resumida en la figura 10. A causa de estos inconvenientes se opto por realizar el desarrollo y las pruebas en un cuarto de 3.50 metros por 6 metros el cual se acerca a las dimensiones de un salón de clases pequeño promedio ademas de tener condiciones de iluminación similares. Respecto al número de personas, participaran 6 individuos con características faciales, estaturas y edades diferentes (3 hombres y 3 mujeres) con el fin de comprobar el funcionamiento del sistema bajo múltiples variables de entrada.

En un ambiente real de salón de clases, nos encontraríamos con inconvenientes como por ejemplo, las personas que están sentadas con una mesa delante de ellas lo cual dificulta en gran medida la correcta identificación de las personas. Por otro lado entre más objetos halla en la escena, se incrementa la posibilidad de presentarse falsos positivos en el procesos de identificación.

4.2. SELECCIÓN DE LA CÁMARA

La primera parte del proyecto consiste en la elección de las herramientas que se van a usar para solucionar el problema planteado. Una parte muy importante del proyecto es la cámara con la que se van a capturar las imágenes y es por esto que su elección debe estar regida por una serie de especificaciones que fueron determinadas por medio de un estudio preliminar de las condiciones del proyecto en el cual se tuvo en cuenta las condiciones ambientales del entorno de prueba, los objetivos planteados en la justificación y el presupuesto asignado al proyecto. Las especificaciones resultantes se listan a continuación:

- El sistema debe funcionar de manera óptima en ambientes no controlados por lo que es necesario que la cámara tenga algún tipo de inmunidad a las diferentes condiciones de iluminación.
- La cámara debería ser fácil de conseguir y de bajo costo.
- Como uno de los objetivos del proyecto es reconocer el foco de atención visual de múltiples personas, se espera que la parte de la imagen que representa la cabeza

de cada persona sea de una resolución pequeña. Es por esto que la resolución de la cámara debería ser mayor o igual a 640x480 pixeles para asegurar que las imágenes segmentadas tengan una tamaño suficiente para poder hacer un reconocimiento con el mínimo de errores.

Se realizó una búsqueda de dispositivos que cumplieran con las condiciones mencionadas anteriormente, cuyo resultado fueron tres cámaras que satisfacían dichos requerimientos: el *Carmine 1.08* de PrimeSense(http://www.primesense.com/solutions/sensor/), el *Xtion PRO LIVE*(http://www.asus.com/Multimedia/Xtion_PRO_LIVE/) de Asus, y el *Kinect* de Microsoft(http://www.microsoft.com/en-us/kinectforwindows/)(figura 12).

Figura 11: Carmine 1.08 de Primesense, Xtion PRO LIVE de ASUS, y Kinect de Microsoft.

Aunque los 3 dispositivos contaban con características muy similares, el Kinect de Microsoft fue el elegido como cámara a usar en el proyecto por su extensa documentación, bajo precio y las herramientas que Microsoft provee para el desarrollo de aplicaciones.

4.2.1. Limitaciones del Kinect

Como se describió anteriormente en la sección 3.2., el Kinect tiene dos tipos de cámara, la cámara RGB o de color y la cámara de profundidad. Esta última es la que será usada para realizar el proceso de identificación del foco de atención visual gracias a su inmunidad a las condiciones variantes de iluminación, al punto de poder funcionar sin ningún problema en situaciones donde la iluminación es nula. Las limitaciones de la cámara de profundidad se encuentran en su rango de funcionamiento el cual se encuentra entre los **800mm** y los **4000mm** para el Kinect que viene con el XBox 360 (el cual es usado en este proyecto) y de **500mm** hasta **3000mm** en el Kinect para Windows. Cualquier objeto que se encuentre fuera de estos rangos tanto por debajo como por encima no es reconocido y es representado como pixeles desconocidos. Por otro lado, el rastreo de esqueleto solo puede ser ejecutado simultáneamente para máximo 2 personas por limitaciones en el Kinect SDK lo cual implica que la estimación del foco de atención visual solo se podrá realizar para dos personas al mismo tiempo.

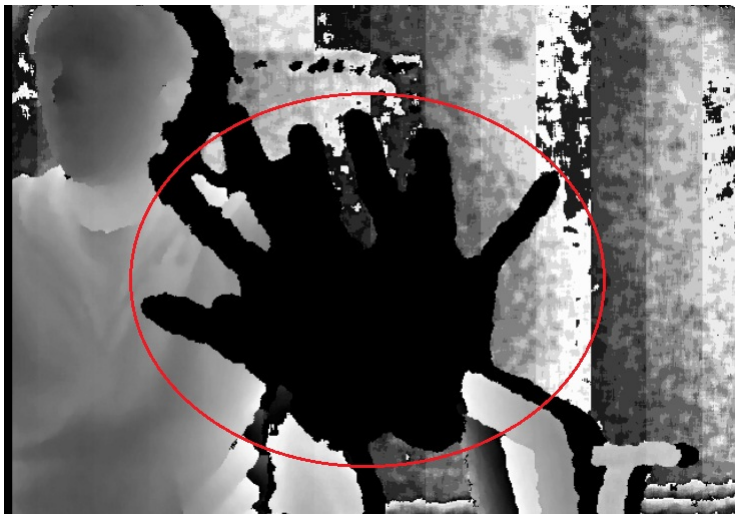

Figura 12: Si los objetos se encuentran muy cerca o muy lejos del kinect, la imagen se degrada como podemos observar en el circulo rojo.

4.3. CAPTURA DE IMÁGENES

Las imágenes de profundidad entregadas por el kinect son diferentes a las típicas imágenes de color de las cámaras convencionales. En una cámara de color, una imagen es representada por una matriz de MxNx3 donde cada pixel se forma por la combinación de 3 valores que representan el valor en rojo, verde y azul. En una imagen de 8 bits de profundidad, cada valor esta en el rango de entre 0 y 255 por lo que en total cada pixel de una imagen RGB utiliza 24 bits para su representación (8 bits por 3 valores).

La imagen de profundidad se representa como una matriz en la que cada pixel contiene información de la distancia a la que se encuentra un objeto en ese punto. Cada pixel utiliza **13 bits** para su representación por lo que las distancias que puede entregar el kinect van desde los 0mm hasta los 8192mm ($2^{13} = 8192$), sin embargo, esto no ocurre en la realidad donde su rango de detección, medido experimentalmente, es de 800mm hasta 3000mm aproximadamente.

Para la representación de estas imágenes el kinect SDK utiliza un tipo de dato llamado *entero corto* o *short integer*. Mientras un entero normal utiliza 32 bits para su representación, un entero corto utiliza 16 bits el cual se acomoda mejor a los 13 bits usados para representar la distancia en la imagen de profundidad. Los 3 bits que sobran son usados para identificar si el pixel corresponde a una persona o si es parte de la escena. Esto quiere decir que el kinect puede encontrar hasta a 6 personas en la escena pero solo puede llevar a cabo rastreo de esqueleto para dos personas al mismo tiempo.

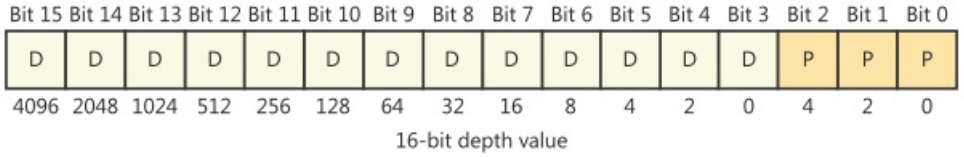

Figura 13: Los 3 bits menos significativos representan si el pixel corresponde a una persona en la escena. Los 13 bits restantes representan la distancia de este pixel medido con respecto al kinect. Imagen extraída desde [11]

Para poder visualizar la información de profundidad que nos entrega el kinect es nece-

sario quitar de cada pixel la información del usuario para que solo nos quede la distancia y convertir el vector resultante en una imagen de color que sea una representación de la profundidad a la que se encuentran los objetos con respecto al kinect.

Con el fin de descartar la información de los 3 bits menos significativos utilizamos la siguiente instrucción en C#:

```
int profundidad = datosProfundidad[indice] >> 3;
```

La variable **datosProfundidad** representa el vector que nos entrega el kinect el cual contiene la información de profundidad y de personas encontradas. El operador >> desplaza hacia la derecha un vector el número de posiciones que se especifique. **indice** es el contador que permite desplazarse a través del vector.

Una vez que el vector contiene solamente la profundidad de cada uno de los pixeles, es necesario convertirlo en una imagen para poder tener una representación visual de la profundidad. Para lograr esto, debemos crear una función cuya salida se encuentre en un rango de 8 bits para cualquier valor de profundidad entregado por el kinect. Una distancia menor o igual a 800 corresponde a un 255 y una distancia mayor o igual a 4000 corresponde a 0. Esto quiere decir, que representando la profundidad en una imagen en escala de grises, los objetos más cercanos a la cámara serán blancos y los mas lejanos serán de color negro.

La función que necesitamos esta construida por tramos de la siguiente manera:

$$f(x) = \begin{cases} 255 & x <= 800 \\ g(x) & 800 \leq x \leq 4000 \\ 0 & x >= 4000 \end{cases}$$

Como $g(x)$ se comporta de manera lineal podemos hallar la pendiente de la recta con los puntos que tenemos.

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$= \frac{0 - 255}{4000 - 800}$$

$$= \frac{-255}{3200}$$

$$= \frac{-51}{640} \tag{1}$$

Si reemplazamos m y dos puntos conocidos de x y y en la ecuación de la recta ($y = mx + b$) podemos hallar el valor de b de la siguiente manera.

$$y = mx + b$$

$$(0) = \frac{-51}{640}(4000) + b$$

$$b = 318.75 \tag{2}$$

De esta manera nuestra función definida por partes queda de la siguiente forma.

$$f(x) = \begin{cases} 255 & x \leq 800 \\ \frac{-51}{640}x + 318.75 & 800 \leq x \leq 4000 \\ 0 & x \geq 4000 \end{cases} \tag{3}$$

La gráfica 14 muestra el comportamiento de la función para cualquier valor de profundidad entregado por el kinect.

Al pasar el vector de profundidad entregado por el kinect por la función que acabamos de crear obtenemos una imagen como la de la figura 15.

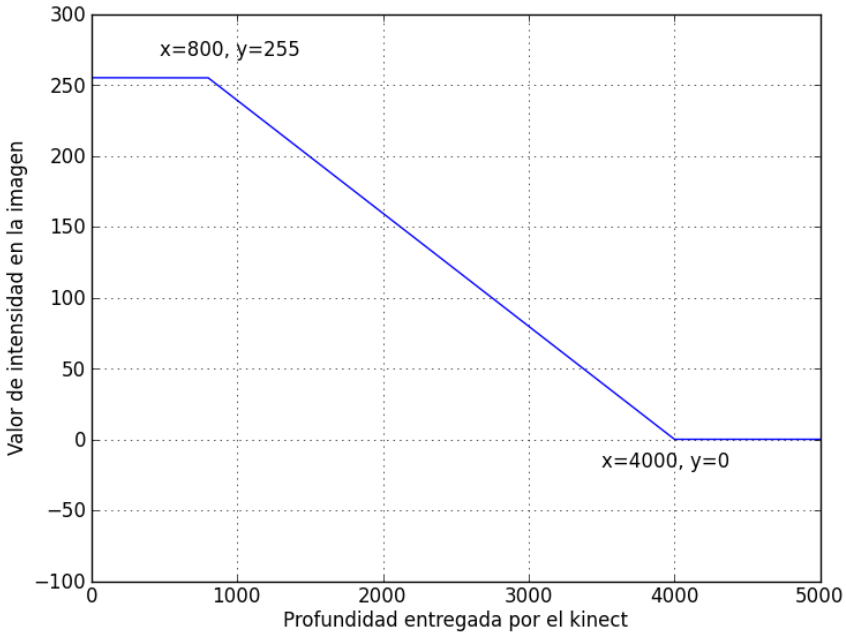

Figura 14: Gráfica de profundidad contra valor de gris en la imagen de salida.

4.4. ENCONTRAR PERSONAS EN LA ESCENA

Como se mostró en la sección anterior, el vector de profundidad entregado por el kinect contiene información sobre las personas que se encuentran en una escena. Encontrar a todas las personas que se en encuentran en una imagen digital se lleva a cabo por un proceso conocido como ***Background removal***. El kinect SDK, se encarga de remover el fondo y entregar la información sobre cuáles pixeles en la imagen de profundidad son parte del fondo y cuáles son personas.

Para llevar a cabo la sustracción de fondo debemos observar los 3 bits menos significativos del vector de profundidad y si su valor es cero quiere decir que este pixel no corresponde a ninguna persona. Es importante tener en cuenta que para que el vector de profundidad contenga la información de si un pixel corresponde o no a una persona, es necesario activar el rastreo de esqueleto en la parte inicial del programa ya que el hardware del kinect por si sólo no diferencia entre personas y fondo.

Figura 15: Imagen resultante de pasar el vector de profundidad entregado por el kinect, por la función de la ecuación 3.

Una vez descartados los pixeles que no pertenecen a una persona obtenemos imágenes como las de la figura 16.

Podemos observar que entre más lejos este una persona del kinect, sus valores de grises van a ser más bajos y entre más cerca se encuentre van a ser más altos. Aunque este comportamiento es el esperado por como esta planteada la función 3, no es lo que buscamos ya que al momento de analizar las cabezas, si una persona está a dos metros y esta misma persona se mueve un metro hacia adelante, aunque su cabeza no halla girado las imágenes serán totalmente diferentes.

En la figura 17 podemos observar como al momento de que una persona se desplaza un metro hacia adelante, la distribución del histograma también se mueve. Es de esperarse que la forma del histograma cambie, pero la distribución debe ubicarse en la misma zona con el fin de buscar patrones que permitan encontrar el angulo de la cabeza.

Para lograr que la imagen de profundidad de las personas sea uniforme sin importar la distancia a la que se encuentren es necesario modificar la función que habíamos creado con anterioridad. En vez de definir que la imagen tenga un valor de 255 para una

Figura 16: Sustracción del fondo.

profundidad de 800mm en todos lo casos, vamos a poner que el pixel con la distancia más cercana de una persona al kinect sea el valor que corresponda a 255 y el valor más lejano sera un valor escogido, de la misma manera, en base al punto mas cercano.

Por ejemplo, si el punto más cercano de una persona al kinect es de 1200mm, el punto mas lejano sera 1200mm + 400mm. Por lo tanto el nuevo rango a convertir sería con el mínimo en 1200 y el máximo en 1600. Si la persona se mueve un metro hacia atrás el nuevo rango seria 2200 y 2600, lo cual asegura que sin importar la distancia a la que se encuentre la persona, su distribución de niveles de grises en la imagen serán similares en la misma posición.

Para hallar la nueva función debemos incluir dos variables que antes eran constantes. Estas dos variables son ***Lejano*** y ***Cercano***. De la misma manera que fue llevado a cabo anteriormente, hallamos la pendiente de la nueva función.

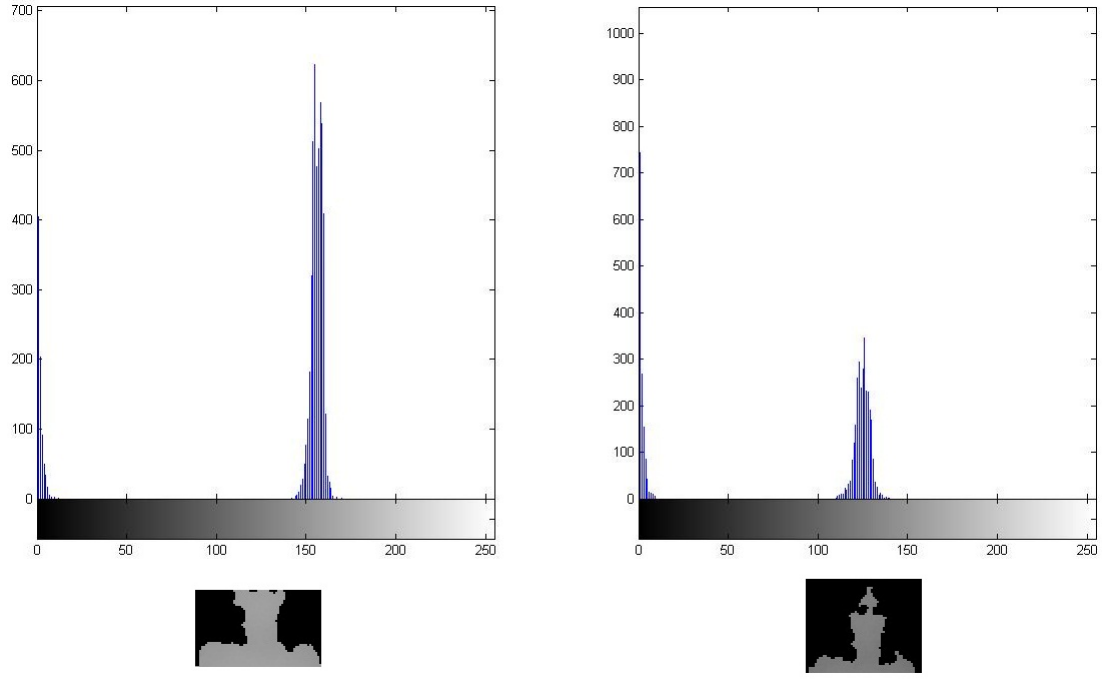

Figura 17: Histograma de una imagen de la cabeza de una persona a 2 metros y 3 metros respectivamente.

$$
\begin{aligned}
m &= \frac{y_2 - y_1}{x_2 - x_1} \\
&= \frac{0 - 255}{lejano - cercano} \\
&= \frac{-255}{lejano - cercano}
\end{aligned} \tag{4}
$$

Después de hallar la pendiente, hallamos el valor de b para completar la ecuación de la recta.

$$y = mx + b$$

$$(0) = \frac{-255}{lejano - cercano}(lejano) + b$$

$$b = \frac{255 * lejano}{lejano - cercano} \tag{5}$$

De esta manera la función que nos permite normalizar las imágenes de profundidad sin importar la distancia queda definida de la siguiente manera:

$$f(x) = \begin{cases} 255 & x \leq 800 \\ \frac{-255}{lejano-cercano}x + \frac{255*lejano}{lejano-cercano} & 800 \leq x \leq 4000 \\ 0 & x \geq 4000 \end{cases} \tag{6}$$

En la figura 18 podemos observar como los histogramas de las imágenes que usan la nueva función son mas parecidos para la misma posición de la cabeza. Además, los detalles faciales son más evidentes ya que el rango a tratar es más pequeño.

En la figura 19 vemos como la nueva función mejora los detalles de la cabeza y hace que la imagen de profundidad de la persona sea constante sin importar la distancia.

Para que la segmentación funcione para dos persona al mismo tiempo debemos hallar el identificador que el Kinect SDK le asigna automáticamente a cada persona encontrada en la escena y enlazar cada identificador con una función en el programa. De esta manera aseguramos que las dos personas tienen una segmentación consistente sin importar la distancia a la que se encuentren del sensor ni tampoco se ve afectada por la posición de otra persona en la escena.

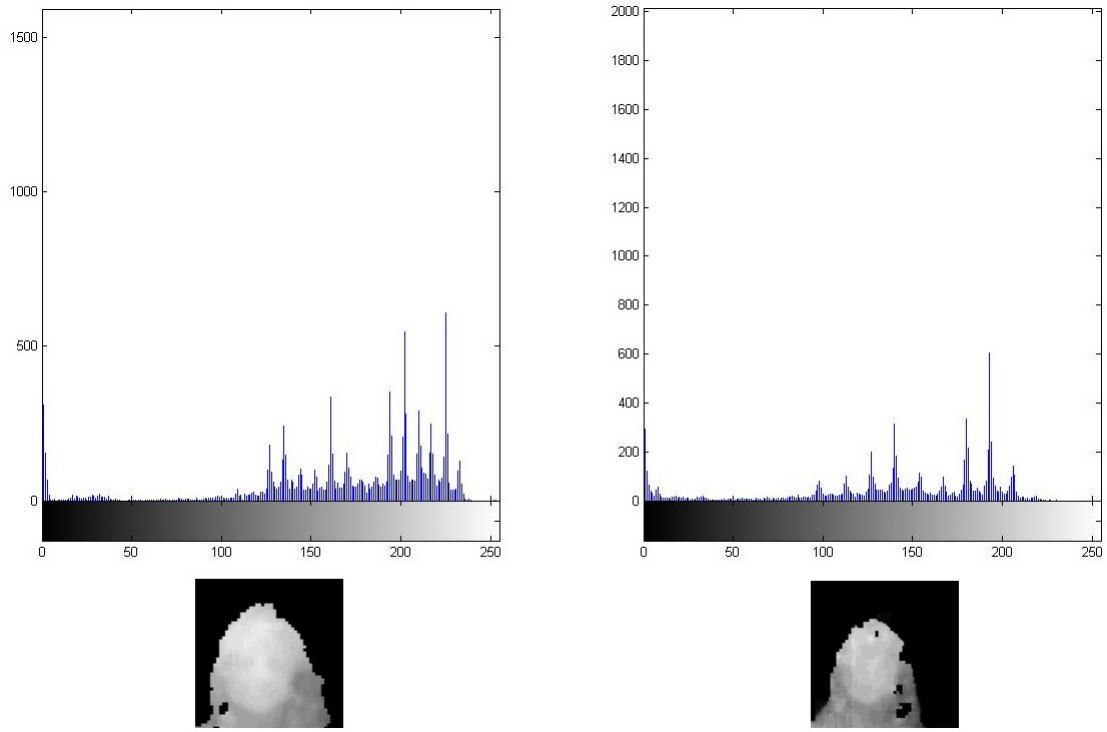

Figura 18: Histograma de una imagen normalizada de la cabeza de una persona a 2 metros y 3 metros respectivamente.

4.5. ENCONTRAR LA CABEZA DE LAS PERSONAS

Una vez que las imágenes de las personas han sido identificadas, normalizadas y separadas del fondo, el siguiente paso es encontrar en la imagen de profundidad completa las partes que representan las cabezas de las personas. Para lograr esto, utilizamos el rastreo de esqueletos que provee el Kinect SDK y el cual nos permite identificar hasta 20 puntos que representan partes del cuerpo [12] los cuales se muestran en la figura 20.

El proceso llevado a cabo para hallar la cabeza de las personas en una imagen de profundidad comienza con encontrar las coordenadas del punto que representa la cabeza y el cual es entregado por el *Skeleton Tracking* del Kinect SDK. Una vez se tienen las coordenadas de la cabeza en la imagen de profundidad procedemos a seleccionar una sección de la imagen de 100x100 píxeles cuyo centro es la coordenada de la cabeza.

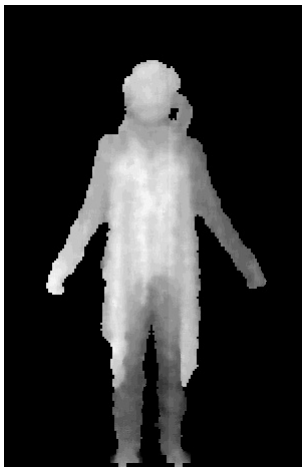

Figura 19: Sustracción de fondo con imagen normalizada.

La medida de las regiones de interés fue seleccionada experimentalmente teniendo en cuenta que fue el tamaño mas pequeño de imagen que permitía visualizar correctamente las cabezas en los rangos de distancias medidos (1m - 3m)(Véase la figura 22).

4.6. DETERMINAR EL ÁNGULO DEL CUERPO

De la misma manera en la que nos hemos estado refiriendo al termino "ángulo de la cabeza" como al grado de inclinación de la cabeza medido con respecto a su eje vertical o comúnmente conocido como *ángulo de guiñada*, podemos hablar del *ángulo del cuerpo* como la rotación de los hombros medido con respecto a su eje vertical. Saber el ángulo del cuerpo nos permite realizar una primera estimación teniendo en cuenta que mientras el cuerpo esté estático la cabeza humana tiene un rango de giro horizontal cercano a los 180 grados[13].

Para hallar el ángulo de un cuerpo haremos uso de la posición de los hombros de la persona. Primero hallamos la distancia de los dos hombros con respecto al sensor tal y como se muestra en la figura **??**. Por ejemplo, si modelamos al cuerpo como una linea recta entre un hombro y otro, el ángulo del cuerpo es igual a cero con respecto al kinect

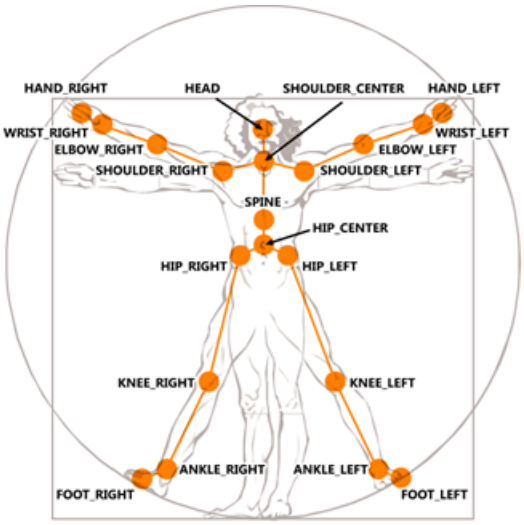

Figura 20: Puntos identificados del cuerpo por el Kinect SDK.

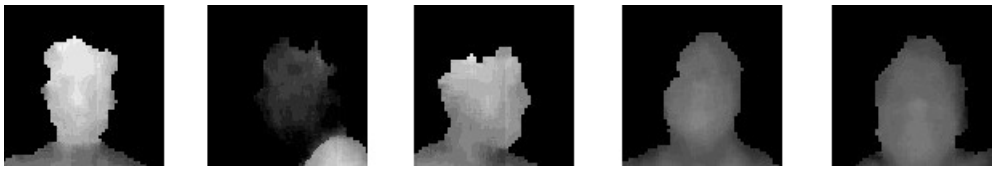

Figura 21: Imágenes de cabezas encontradas en una imagen de profundidad.

como se aprecia en la figura 24.

Cuando el cuerpo ya no es paralelo al kinect, es necesario encontrar una formula que nos permita describir el ángulo del cuerpo en cualquier momento. De la figura 25 podemos definir lo siguiente:

$$Sin(\alpha) = \left(\frac{X_1 - X_2}{X_3}\right)$$

$$\alpha = Sin^{-1}\left(\frac{X_1 - X_2}{X_3}\right) \tag{7}$$

El valor de X_3 es hallado previamente mediante la resta de las coordenadas en X del hombro izquierdo y el derecho (este valor también es entregado por el Kinect SDK y su

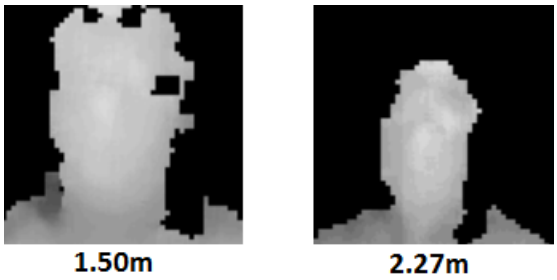

Figura 22: Imagen de la cabeza de la misma persona a 1.50m y a 2.27m.

Skeleton tracking).

Aplicando la formula 7 podemos realizar una primera estimación del ángulo del cuerpo, y por consiguiente de la cabeza, con respecto al Kinect. El resultado de hallar el ángulo para una personas se puede observar en la figura 26 y para dos personas en la figura 27.

Aunque este enfoque arroja una muy buena aproximación del ángulo del cuerpo, solo funciona mientras las personas se encuentren de frente al kinect por lo que su rango de reconocimiento esta entre los -90 y los +90 grados medidos con respecto al kinect.

4.7. DETERMINAR EL ÁNGULO DE LAS CABEZAS

Existen diferentes métodos que permiten llevar a cabo una estimación de la posición de la cabeza[14]. Cada método tiene una relación de costo beneficio la cual debemos tener en cuenta al momento de seleccionar alguna en particular para ser usada en alguna aplicación. El método escogido para hallar el ángulo de guiñada de las cabezas en una escena es el *Appearance Template* o *Plantillas de apariencia*. Este método consiste en comparar una imagen de la cual no se sabe a que ángulo pertenece y compararla con imágenes plantillas previamente seleccionadas y etiquetadas.

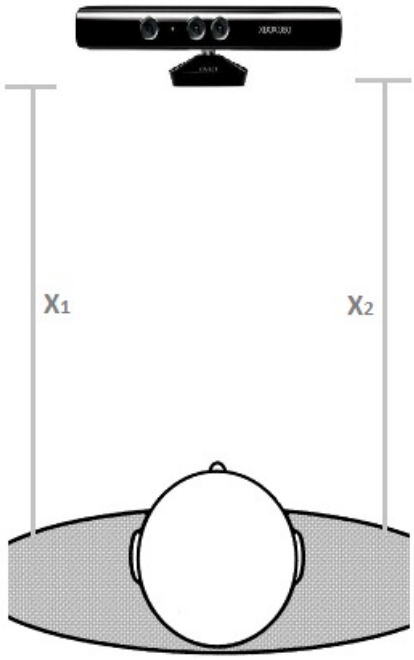

Figura 23: Distancia de los hombros con respecto al sensor.

4.8. EXPORTAR IMÁGENES

Por limitaciones de hardware no fue posible hallar el ángulo de la cabeza en tiempo real, por lo que el programa realizado exporta imágenes de cabezas etiquetas según el angulo al que corresponden. Por cada ángulo de los planteados en los objetivos (0, 45, 90, 135, 180, 225, 270, 315) se crea una carpeta y en ella se ponen todas las imágenes plantillas que correspondan a ese ángulo.

4.9. PROCESAMIENTO CON MATLAB

Cuando se quiere hallar el ángulo al que pertenece una imagen desconocida la pasamos por un script hecho en Matlab el cual compara la imagen con todas las imágenes de cada carpeta. Cuando se comparan las imágenes se hace por medio de la función

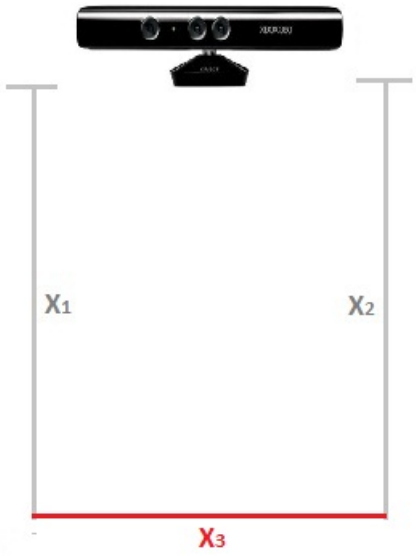

Figura 24: Representación de la distancia entre hombros.

corr2 la cual devuelve un coeficiente de correlación que tiende a 1 cuando las imágenes son muy similares o tiende a 0 cuando son muy diferente[15]. Para determinar si una imagen pertenece a un ángulo en particular, se suman todos los coeficientes de correlación resultantes de comparar la imagen con todas las plantillas de una carpeta y por ultimo, este resultado se divide entre la cantidad de imágenes plantilla que halla en la carpeta dando como resultado el promedio de coeficientes de correlación. Por medio de las pruebas realizadas se pudo llegar a la conclusión de que para que una imagen pertenezca a un ángulo en particular, su promedio de coeficientes de correlación debe ser superior a 0.6. El método utilizado para hallar el angulo de las cabezas tiene entre sus ventajas la facilidad de implementación y resultados positivos en un 90% de las imágenes comparadas. Entre sus desventajas podemos encontrar que su velocidad de procesamiento es bastante lenta ya que compara una a una todas las imágenes. Ademas, para lograr una buena exactitud, los datasets deben ser muy grandes comparados con otros métodos.

En las pruebas realizadas para el proyecto se utilizaron 1600 imágenes como plantillas (8 categorías de 200 imágenes cada una) y se obtuvo exactitud del 95% en 20 pruebas realizadas (algunos resultados se pueden apreciar de la figura 28 a la 31).

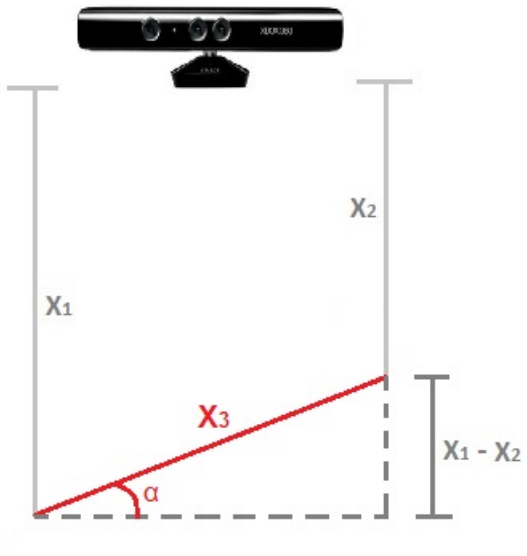

Figura 25: Planteamiento de ecuaciones para hallar el angulo del cuerpo.

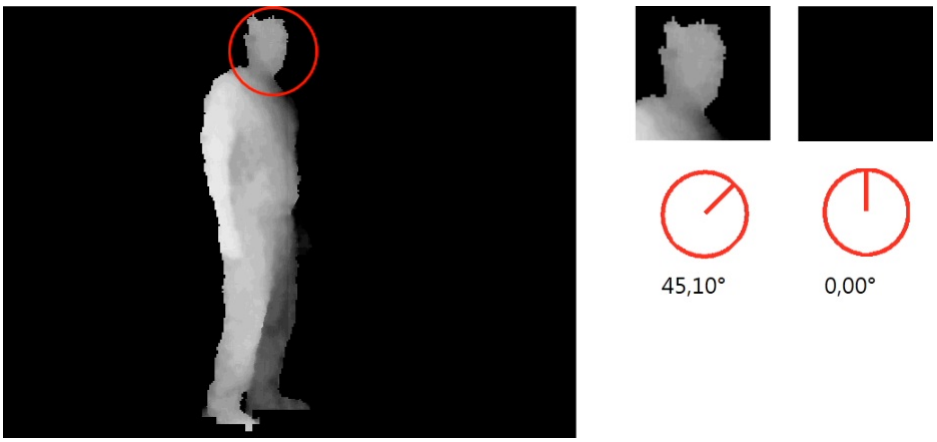

Figura 26: Herramienta hallando en tiempo real el ángulo del cuerpo.

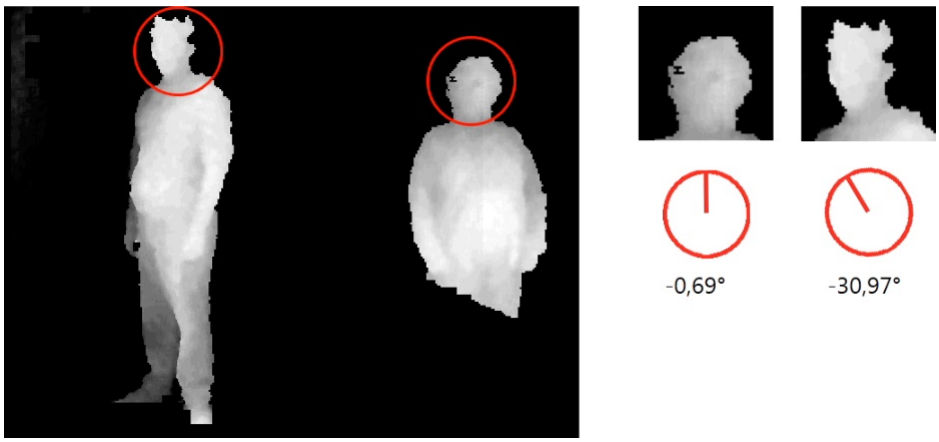

Figura 27: Herramienta hallando en tiempo real el ángulo del cuerpo para dos personas.

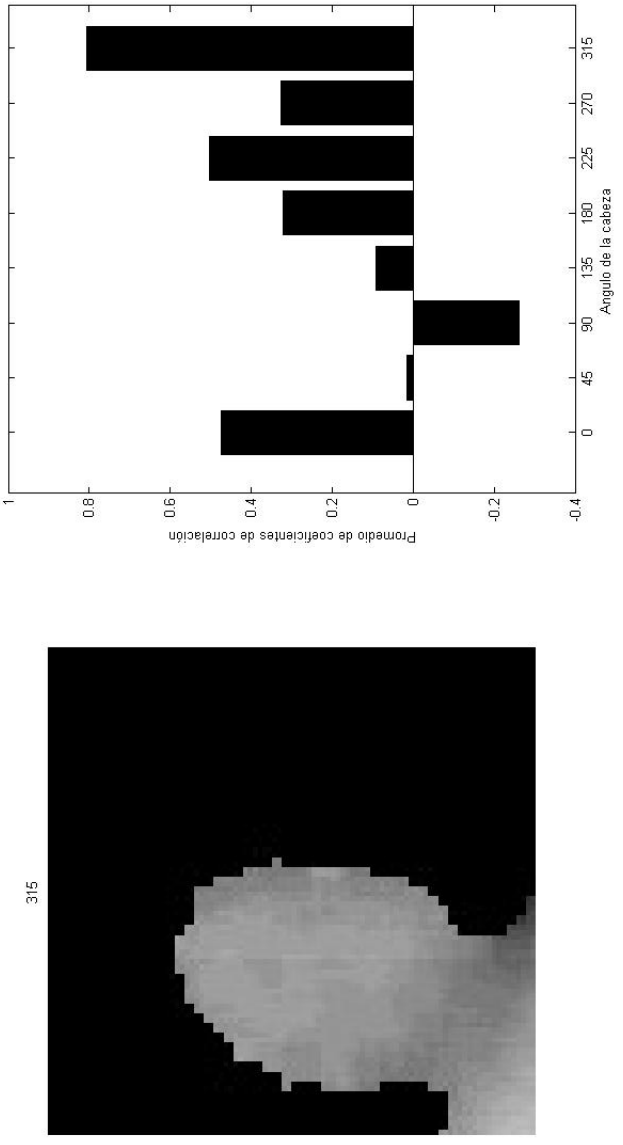

Figura 28: Verdadero positivo 315 grados.

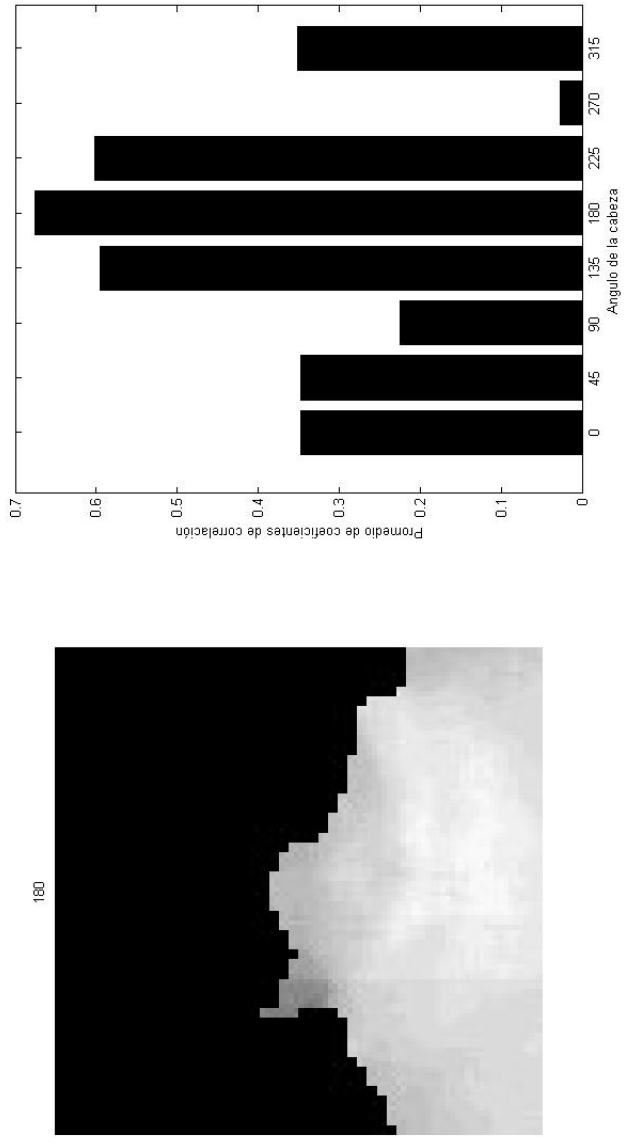

Figura 29: Verdadero positivo 180 grados.

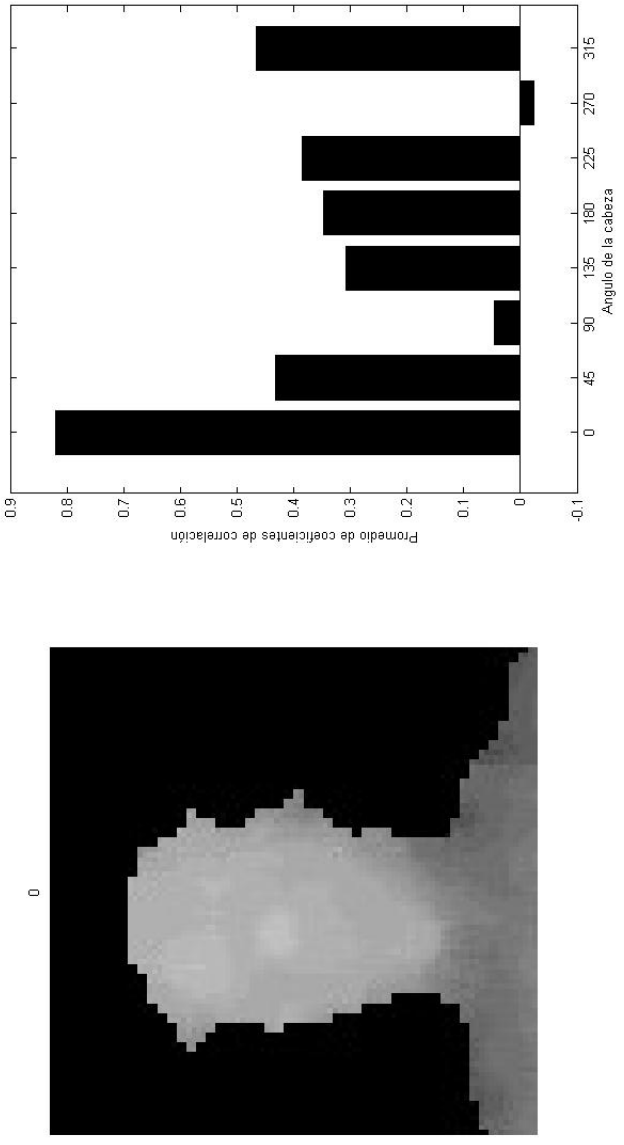

Figura 30: Verdadero positivo 0 grados.

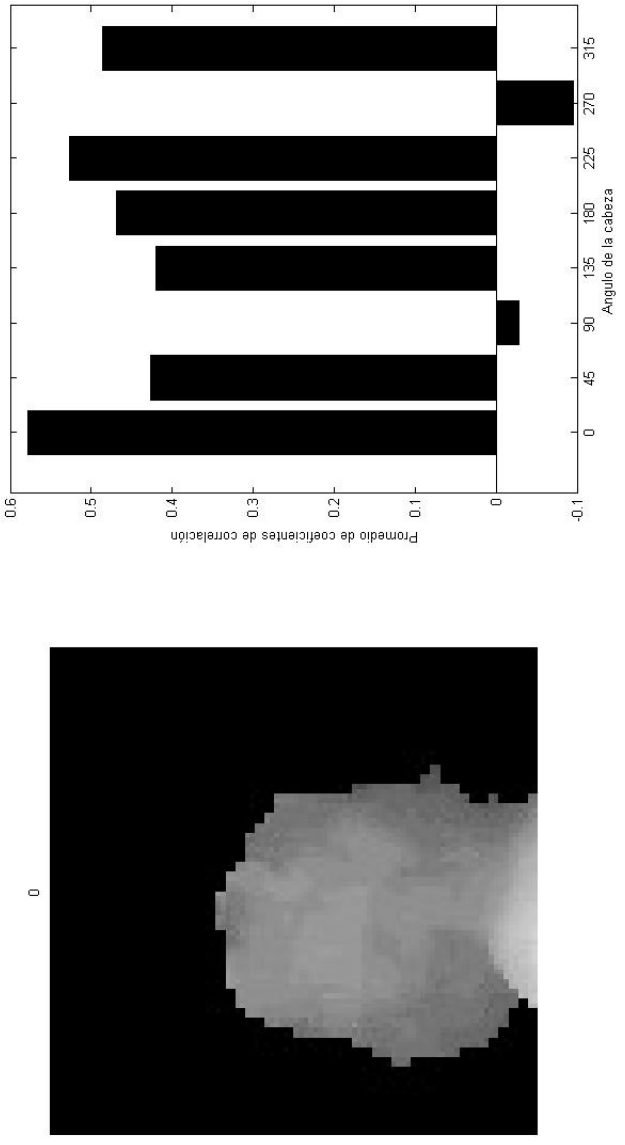

Figura 31: Falso positivo 0 grados.

5. PRUEBAS

En el presente trabajo se desarrollo una herramienta que permite estimar el foco de atención de 2 personas en un ambiente de simulación controlado. En primera medida, se realizo una etapa de elección de la cámara que seria el eje central del proyecto. Una vez seleccionado el Kinect de Microsoft, se diseño una función que eliminara el fondo en una imagen de profundidad y que ademas normalizara la parte de la imagen para que las personas que se encontraban en la escena tuvieran unos valores de grises sin importar la distancia a la que se encontrara. De esta manera se lograban aprovechar mejor los detalles faciales de las personas que pueden dar pistas del ángulo de su cabeza sin depender de la distancia a la que se encuentre la persona. Luego, y por medio del Kinect SDK, se lograba extraer las coordenadas de las cabezas de las personas que se encontraban en la escena y se hallaba una sub-imagen que contenía única y exclusivamente información de profundidad de la cabeza. Como paso adicional se hallo el ángulo de inclinación horizontal del cuerpo con el fin de ser utilizado como información adicional en la siguiente y última etapa. Por ultimo y con un script externo al programa principal, comparamos imágenes nuevas con datasets previamente creados y guardados para lograr una estimación del ángulo de guiñada.

5.1. DESEMPEÑO

La herramienta que segmenta y almacena las imágenes de las cabezas de las personas tiene una efectividad de detección del 100% cuando el angulo inicial del cuerpo es de 0 grados y su efectividad disminuye cuando el cuerpo comienza a girar llegando a ser del 10% aproximadamente cuando el angulo inicial del cuerpo es mayor a 90 grados y menor a 270 grados. Una vez el cuerpo es detectado y el programa comienza a almacenar las

imágenes el sistema se comporta de una manera altamente estable.

En la etapa de captura de imágenes se utilizo un equipo con 2GB de RAM y un procesador de 2.4GHz de 32 bits de un solo núcleo con Windows 7 el cual respondía de una manera aceptable teniendo muy pocos cuelgues en el desempeño por lo que se puede concluir que el sistema funciona de una manera bastante eficiente con relativamente pocos recursos. El sistema no se ralentiza de manera significativa cuando detecta a dos personas al mismo tiempo por lo que se puede afirmar que en futuras versiones del Kinect SDK es posible que aumenten el número de personas que pueden ser rastreadas al mismo tiempo.

El lugar de simulaciones donde se realizaron las pruebas demostró el correcto funcionamiento del sistema frente a condiciones impredecibles de luz llegando al punto de funcionar sin ninguna diferencia cuando había buena iluminación y cuando el ambiente estaba en total oscuridad.

El script de Matlab, aunque lento, tiene un alto indice de clasificaciones de imágenes correctas dentro de las categorías de ángulos disponibles.

5.2. TRABAJO FUTURO

La principal linea de trabajo para futuro será la implementación completa del sistema en tiempo real permitiendo integrar la herramienta con otros programas más complejos que requieren como entrada el foco de atención visual de las personas. También se tiene como trabajo futuro planteado reducir cada vez más el rango de detección de los ángulos que detecta el sistema, pasando de rangos de 45 grados hasta llegar a 1 grado. El algoritmo diseñado para normalizar la imagen de profundidad sirve como punto de partida para otro tipo de proyectos en los que la segmentación de partes del cuerpo se realizaba usando imágenes en imágenes de color dando u paso hacia adelante en el desarrollo de aplicaciones de visión artificial. Para asegurar la continuación y el aprovechamiento de los algoritmos desarrollados en este proyecto, todo el código y el documento estarán disponibles en un repositorio de Github(http://www.github.

com/raerpo) bajo una licencia GPL la cual protege la propiedad intelectual del código ademas de asegurar que este pueda ser estudiado, modificado, compartido y usado por cualquiera que así lo desee siempre y cuando el código resultante mantenga la misma licencia.[16]

5.3. DEFICIENCIAS

La principal deficiencia del proyecto es la imposibilidad de realizar la detección del ángulo de las cabeza en tiempo real. Por otro lado, los rangos de detección son demasiado grandes aunque cumplen con los estipulados en los objetivos.

6. CONCLUSIONES

- Si bien la herramienta es capaz de estimar la posición de la cabeza con base en la información del ángulo del cuerpo y externamente con el script realizado en Matlab, mientras no sea capaz de realizar los cálculos de la estimación del foco de atención visual en tiempo real, no podrá ser integrada con otro tipo de programas.
- La estimación del foco de atención visual o para este caso en particular, el ángulo de la cabeza con respecto al kinect, teniendo como enfoque el método utilizado en las pruebas (Correlación de matrices) puede llegar a ser lento, pero es bastante efectivo con una gran cantidad de datasets de profundidad.
- Trabajar con imágenes de profundidad soluciona muchos de los problemas a los que se enfrenta cualquier sistema de visión artificial enfocado en los comportamientos. Una unión de la imagen de profundidad con la imagen de color entregada por el kinect sirve como punto de partida de muchos futuros proyectos.
- Aunque es posible identificar comportamientos de una persona estimando su foco de atención visual, la herramienta aún no es lo suficientemente robusta como para identificar comportamientos en un ambiente sin ningún tipo de control. Aún así, el algoritmo diseñado para la normalización de una imagen de profundidad y la segmentación de la mismas según partes del cuerpo permitiría dar un paso adelante en la creación de más y mejores herramientas.
- La forma en que se clasificaban las imágenes en los diferentes rangos planteados resulto ser bastante eficaz y sencilla de implementar. El éxito de un método tan simple consiste en la buena segmentación y calidad de las imágenes aportadas por la herramientas desarrollada con el Kinect SDK.
- El potencial de herramientas como el Kinect en aplicaciones de visión artificial son innumerables por lo que es importante estar al tanto de los adelantos en este tipo de tecnologías que impulsan la creatividad de las personas en todas las áreas.

BIBLIOGRAFÍA

[1] 'Kinect for Xbox 360' is Official Name of Microsoft's Controller-Free Game Device, http://goo.gl/yMK0u, 2010.

[2] D. Gantenbein, "Kinect Launches a Surgical Revolution", http://goo.gl/uyIdT, 2012.

[3] S. Langton, H. Honeyman, y E. Tessler, "The influence of head contour and nose angle on the perception of eye-gaze direction" Perception and Psychophysics, vol. 66, no. 5, pag. 752–771, 2004.

[4] B. Benfold, I. Reid, Colour Invariant Head Pose Classification in Low Resolution Video, 2008.

[5] A. Launnila, Real-Time Head Pose Estimation in Low-Resolution Football Footage, School of Computer Science and Engineering Royal Institute of Technology , 2009.

[6] E. Seemann, K. Nickel, and R. Stiefelhagen, "Head pose estimation using stereo vision for human-robot interaction," 2004.

[7] J. Orozco, S. Gong, T. Xiang. "Head Pose Classification in Crowded Scenes", 2009.

[8] S. Jobs. "iPhone Keynote", http://goo.gl/jFqtF, 2007.

[9] R. Bolt, "Put-that-there": Voice and Gesture at the Graphics Interface, http://goo.gl/wCQwn

[10] A. Jana, "Kinect for windows SDK Programming Guide", 2012.

[11] R. Miles, "Learn the kinect API", 2012.

[12] MSDN España, "Reto SDK de Kinect: Detectar posturas con Skeletal tracking", 2011.

[13] V. F. Ferrario, C. Sforza, G. Serrao, G. Grassi, and E. Mossi, "Active range of motion of the head and cervical spine: a three-dimensional investigation in healthy young adults", 2002.

[14] E. Murphy, M. Manubhai, Head pose estimation in computer vision: A survey.

[15] 2-D Correlation coefficient, http://www.mathworks.com/help/images/ref/corr2.html.

[16] GNU General Public License. http://www.gnu.org/licenses/gpl.html