Universidad Zaragoza
1542

Trabajo Fin de Grado

Detección de personas para simulación de prótesis visual

Autor

Manuel Guerrero Viu

Director

Jesús Bermúdez Cameo

Ponente

José Jesús Guerrero Campo

Grado en Ingeniería Electrónica y Automática

ESCUELA DE INGENIERIA Y ARQUITECTURA
2016

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

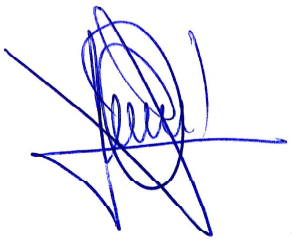
D./Dª. MANUEL GUERRERO VIU,

con nº de DNI 73160786-V en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) GRADO INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA, (Título del Trabajo) DETECCIÓN DE PERSONAS PARA SIMULACIÓN DE PRÓTESIS VISUAL

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 07 de Noviembre de 2016



Fdo: MANUEL GUERRERO VIU

Detección de personas para simulación de prótesis visual

RESUMEN

Las personas reciben la información del entorno que les rodea por medio de los sentidos. La vista es el sentido que más información aporta acerca de una escena y de los objetos ubicados en la misma. A través de la visión y el aprendizaje somos capaces de reconocer objetos, personas, etc, pudiendo así, interactuar con ellos. Sin embargo, algunas patologías pueden causar daños en los sistemas visuales, conduciendo incluso a la ceguera. En función del tipo de daño, existen investigaciones en curso para colocar una prótesis de visión biónica que, mediante una estimulación eléctrica en determinadas zonas del nervio óptico o del cortex cerebral, permiten la visualización de puntos de luz denominados fosfenos. Para la estimulación de los fosfenos, una de las posibilidades es la captura de información de la escena mediante una cámara. Las técnicas actuales de procesamiento de imagen que se aplican sobre las prótesis visuales son bastante limitadas. La introducción de técnicas avanzadas de visión por computador incluyendo información de profundidad puede provocar un punto de inflexión en la forma de interacción con el entorno de las personas operadas con estos novedosos implantes.

En este proyecto se ha avanzado en el desarrollo de un simulador de prótesis visuales considerando nuevas técnicas de visión por computador para favorecer la interpretación del entorno. En particular, se ha desarrollado una aplicación para la detección de personas y su representación mediante fosfenos. Se ha partido de un sistema compuesto de un sensor de profundidad RGB-D, Kinect v2 y de un sistema de realidad virtual, Oculus DK2. Mediante técnicas de visión por computador se ha inferido una descripción articular de la persona y una descripción de la cara que incluye ojos, boca y estado de ánimo. Esta información es representada de manera icónica en el simulador de visión protésica para una mejor interpretación por parte del usuario. Además, se han realizado varias pruebas para poder evaluar diferentes tipos de representación, en función de los distintos mapas de fosfenos diseñados.

Los resultados obtenidos han sido bastante satisfactorios, cumpliendo todas las previsiones realizadas. Uno de los objetivos más importantes ha sido la de realizar la representación mediante fosfenos proponiendo varios ejemplos que permitan una correcta interpretación por parte del usuario pese a disponer de una resolución reducida.

INDICE

Capítulo

1

INTRODUCCIÓN

La vista es el sentido que más información aporta acerca de una escena y de los objetos ubicados en la misma. Según la Organización Mundial de la Salud (OMS) alrededor de 285 millones de personas padecen algún tipo de discapacidad visual, de las cuales 39 millones son ciegas [IAPB-2010]. Esto supone un 0,7% de la población mundial. En España el envejecimiento de la población y el crecimiento de enfermedades crónicas han propiciado que casi 1 millón de personas sufran de ceguera en la actualidad [INE].

Diferentes trabajos de investigación han descubierto que la estimulación eléctrica de la corteza cerebral u otras partes de la vía visual permiten que la persona perciba puntos brillantes de luz llamados fosfenos [BRINDLEY-1968]. Las prótesis visuales consisten en implantes que aplican estimulación eléctrica en la retina, el nervio óptico o la corteza cerebral por medio de un conjunto de electrodos para generar una red de fosfenos similares a una imagen de puntos de baja resolución [DAGNELIE-2011]. Se requiere de un dispositivo que recoja información de la escena, la procese y la transforme en una señal de excitación. El dispositivo de captación de la escena suele ser una cámara convencional, y en su procesamiento es donde entran las técnicas de visión por computador para lograr mostrar al paciente por medio de esos estímulos visuales la información más trascendental del mundo que le rodea.

Por desgracia, la resolución de la rejilla de fosfenos producidos se ve limitada por la biología, la tecnología y la seguridad del propio paciente [DENNIS-2012]. Actualmente se están desarrollando dispositivos que proporcionan unas decenas de fosfenos. Teniendo en cuenta dicha resolución se han realizado importantes esfuerzos en la aplicación de algoritmos de visión para conseguir mejoras [BARNES-2013], aunque la manera de procesar y codificar la información en el dispositivo de manera útil y significativa es todavía una cuestión abierta.

La visión por computador puede hacer un mejor uso de la limitada resolución resaltando características de las imágenes, reconociendo objetos e interpretando la estructura 3D de la escena. La introducción de información de profundidad puede permitir una mejora en la imagen, una mayor robustez en el reconocimiento del espacio y una gran ayuda para la interacción de la persona con el entorno. La

manera de procesar y codificar la información de la imagen en el dispositivo puede llegar a ser muy útil si se hace de una manera correcta.

Todavía queda un largo trabajo en la parte ingenieril para lograr que un paciente ciego con una prótesis visual pueda interpretar su entorno por medio del sentido de la vista. Quizá es este el factor que menos se ha desarrollado en la actualidad, por lo que actualmente es una vía importante de investigación. Utilizar técnicas de visión por computador para procesar e interpretar imágenes de color y profundidad puede contribuir en gran medida a este objetivo.

No menos importantes son los aspectos neurológicos y psicológicos de la visión, puesto que son los caminos que nos indican cómo debemos procesar la información del entorno para que sea interpretada intuitivamente por cualquier persona.

La colaboración con personas con discapacidades visuales supone, sin duda alguna, la opción más real y conveniente. Sin embargo, para poder avanzar de manera ágil en la investigación necesitamos una gran cantidad de información acerca de lo que el paciente es capaz de ver. Por ello surgen los simuladores de prótesis visuales (Simulated Prosthetic Vision, SPV) que emulan el comportamiento de este tipo de prótesis y nos permiten comprender cómo percibiría su entorno una persona privada de la visión. El objetivo de este trabajo es desarrollar un prototipo para la simulación de prótesis visuales que permita investigar en técnicas de codificación de la información del entorno para adaptarla a sistemas de visión biónica. El simulador de prótesis visuales recibe imágenes tridimensionales capturadas mediante un sistema RGB-D y las transmite a un casco de realidad virtual, como se verá más adelante.

Este proyecto surge con el propósito de unir los avances biomédicos en prótesis visuales con una rama de la ingeniería como es la visión por computador. Se trata de un tema de gran impacto social, pretendiendo mejorar la calidad de vida de las personas con discapacidad visual, facilitando su movilidad e interacción con el entorno.

El proyecto se ubica en el área de Ingeniería de Sistemas y Automática de la Universidad de Zaragoza, haciendo uso de la rama de visión por computador. El departamento ha puesto a disposición del proyecto todas las herramientas y dispositivos necesarios, como son cámaras, gafas de realidad virtual, computadores e incluso la ayuda del personal que ha sido necesaria.

1.1 PRECEDENTES

La realización de este proyecto sirve como continuación al Trabajo Fin de Master (TFM) realizado por Alberto Badías Herbera *Simulación de prótesis visual con sensor RGB-D* [BADIAS-2016], y comparte con éste, ser el punto de partida para la investigación en temas relacionados con las prótesis visuales. En el departamento de Informática e Ingeniería de Sistemas ya se dispone de la experiencia necesaria para abordar este tipo de problemas. Los resultados en visión por computador son reconocidos a nivel mundial, en particular las técnicas desarrolladas que capturan información tridimensional a partir de cámaras en movimiento. Desde 2010 se está trabajando en una línea de asistencia personal. Se han usado cámaras omnidireccionales que aportan un campo de visualización mayor, pudiendo captar información de todas direcciones [LÓPEZ-NICOLÁS-2014]. También se ha trabajado con sistemas RGB-D que introducen la profundidad de la escena, mejorando considerablemente la localización de los objetos [GUTIÉRREZ-GÓMEZ-2012] para su reconstrucción [PUIG-2014] y el guiado de las personas con deficiencias visuales [ALADREN-2016].

Se pretende continuar en la línea de asistencia personal pero haciendo hincapié en la representación mediante fosfenos de personas, en especial de caras, una vez detectadas con el sistema de visión por computador. De esta forma, se busca dar un salto cualitativo en el contenido semántico de la información que va a recibir el usuario. Las técnicas y herramientas de percepción tridimensional permiten una representación icónica que facilita la interpretación de la imagen de fosfenos.

Por tanto, el siguiente paso es incidir en la transmisión de la información del entorno al usuario, apoyándose en los sistemas de detección de los que ya se dispone.

1.2 OBJETIVOS Y ALCANCE

A continuación se van a exponer de forma resumida los objetivos concretos de este proyecto:

- **Detección de personas mediante cámara RGB-D:** El primer paso es conocer los diferentes software disponibles para la detección de personas (tanto cara como esqueleto) disponibles en la literatura científica, así como su comprensión y utilidad para llevar a cabo las modificaciones necesarias y poder así adecuarlo a las necesidades de este proyecto.
- **Representación icónica de la detección mediante fosfenos:** El siguiente paso, consiste en realizar el procesamiento de los datos obtenidos mediante el sistema de detección y diseñar los diferentes tipos de representación. Se han propuesto e implementado varios tipos de representación para la cara y cuerpo, con el fin de utilizar uno u otro en función de las necesidades de la aplicación.
- **Experimentación y valoración** de los distintos tipos de representación propuestos, conociendo las ventajas e inconvenientes de cada caso. Se han propuesto además diferentes mapas de fosfenos con distintas resoluciones y se ha mostrado la adecuación de cada tipo de representación a esas resoluciones.
- **Integración de la detección y la representación en un sistema de realidad virtual:** Se ha buscado integrar tanto la detección como la representación en un mismo sistema de realidad virtual debidamente calibrado, obteniendo de esta forma el simulador de prótesis visual que se buscaba desde un principio.

A continuación se va a exponer una breve descripción del contenido de los distintos capítulos que se tratarán más adelante:

- **Capítulo 2:** Descripción y análisis de los dispositivos utilizados (cámara RGB-D y gafas de realidad virtual).
- **Capítulo 3:** Explicación técnica de la implementación realizada en cuanto a la detección del cuerpo y de la cara
- **Capítulo 4:** Descripción de las principales características tanto de las prótesis visuales disponibles en la actualidad, como de los mapas de fosfenos utilizados en el proyecto. Explicación técnica de la implementación realizada en cuanto a representación icónica mediante fosfenos.
- **Capítulo 5:** Propuesta de diferentes tipos de representación icónica describiendo posibles ventajas e inconvenientes de cada una de ellas
- **Capítulo 6:** Recopilación de las conclusiones y opciones de trabajo futuro a realizar.

- **Anexo A:** Calibración de los dispositivos empleados (Kinect v2 y Oculus Rift DK2).
- **Anexo B:** Esquema general del software, donde se expone la estructura principal del mismo.

Capítulo

2

ELECCIÓN DE LOS DISPOSITIVOS UTILIZADOS

Para simular el efecto que produce una prótesis visual sobre personas que han perdido la visión se requiere una cámara que capture la escena y un medio sobre el que se proyecte la matriz de fosfenos con información visual. La cámara elegida es el modelo Kinect en su segunda versión, del fabricante Microsoft, y las gafas de realidad virtual son las Oculus Rift Development Kit 2 de la compañía Oculus VR. Estos dispositivos son idénticos a los empleados en el Trabajo Fin de Master (TFM) realizado por Alberto Badías Herbera *Simulación de prótesis visual con sensor RGB-D* [BADIAS-2016], que es la base sobre la que se apoya este proyecto.

2.1 DISPOSITIVOS EMPLEADOS

2.1.1 KINECT v2

El sensor que captura la escena es el Kinect v2 o Kinect for Xbox One. Este dispositivo permite tomar información de color mediante una cámara convencional (RGB) e información de profundidad haciendo uso de una cámara de infrarrojos y un patrón emisor de luz.

a) b)

Figura 2.1. Kinect v2. a) Dispositivo tal y como se suministra. b) Dispositivo desensamblado para observar su interior

Para capturar la profundidad utiliza una técnica llamada tiempo de vuelo (Time-Of-Flight, ToF) que consiste en estimar la profundidad mediante una medida del tiempo de reflexión de un haz de luz infrarroja. En este caso, la cámara de luz infrarroja (IR) dispone de un filtro para excluir la información de luz asociada al espectro visual, reduciendo la banda de trabajo y haciendo al sensor más robusto ante cambios de iluminación. Conociendo la velocidad de transmisión de la luz en el aire, el tiempo transcurrido entre el instante de emisión del patrón y su lectura es traducido a distancia. Sin embargo, para poder tomar medidas temporales precisas la iluminación de los puntos emitidos no es constante, y aparecen dos vertientes dentro de ToF: directa (pulsos de luz) e indirecta (modulación en amplitud de iluminación). Según se cree, el sensor Kinect está empleando el modo indirecto, ya que Microsoft adquirió una empresa con patentes relacionadas con este tipo concreto de técnicas [INSIDER-2015], [LACHAT-2015]. La distancia estimada d entre el sensor y el objeto capturado depende del desplazamiento de fase entre la señal emitida y la señal recibida ($\Delta\boldsymbol{\phi}$) determinado por la ecuación 2.1 [CASTANEDA-2011]:

$$d = \frac{(\Delta\boldsymbol{\phi})}{4\pi\omega} c \qquad \text{(ecuación 2.1)}$$

Donde $\Delta\boldsymbol{\phi}$ es el desplazamiento de la fase, c es la velocidad de la luz ($\approx 3\ 10^8\ \frac{m}{s}$) y ω es la frecuencia de modulación. Es importante añadir que no es posible manipular la frecuencia de modulación de los parámetros internos de Kinect, esta información no es accesible.

Puesto que no se conoce el valor de la frecuencia de modulación, no se puede estimar la resolución mínima teórica, aunque como se verá más adelante algunos autores han publicado trabajos donde fijan estos valores.

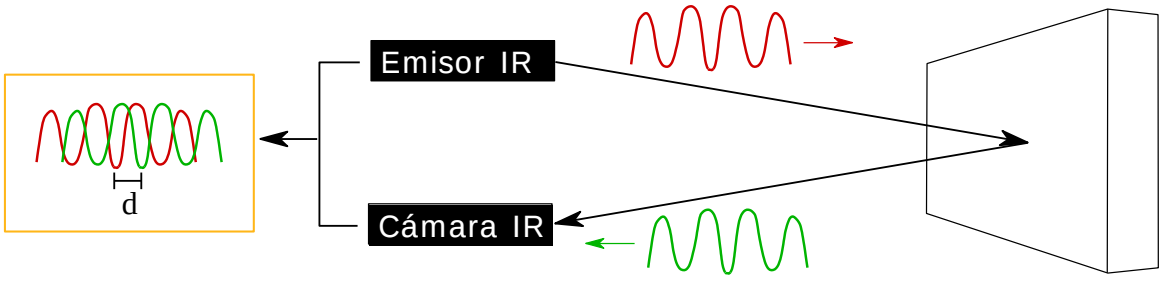


Figura 2.2 Esquema de funcionamiento de la tecnología ToF empleada por Kinect v2

Existen más técnicas para recuperar la profundidad en una escena de manera estática, como por ejemplo el uso de cámaras estéreo (multicámara), o técnicas basadas en luz estructurada (técnica empleada por la primera versión del sensor

Kinect). La primera de ellas requiere el uso de dos (o más) cámaras, que pueden ser de tipo convencional, que capturando un mismo objeto desde posiciones distintas permiten triangular su posición en el espacio basándose en técnicas fotogramétricas. Las técnicas basadas en luz estructurada utilizan esta misma premisa pero sustituyen una de las cámaras por un patrón de luz. Esto es lo que hacía la primera versión de Kinect, proyectando un patrón conocido de luz infrarroja sobre la escena e inspeccionando la distorsión de dicho patrón [GENG-2011].

En el trabajo de Larry Li [LI-2014] aparece una comparativa entre estas tres técnicas. En comparación con la luz estructurada, Time-Of-Flight permite un dispositivo más compacto al no necesitar una cierta distancia entre el emisor y la cámara de infrarrojos para triangular la profundidad (conocido como baseline), es menos sensible a las condiciones de iluminación y tiene un mejor tiempo de respuesta debido a su tecnología. Con respecto a la visión estéreo, el tiempo de vuelo permite emplear una única cámara, de nuevo sin la necesidad de esa baseline que limita el rango de profundidad a capturar, que quizá es la mayor desventaja de la visión estéreo, junto con el problema del reconocimiento de características, puesto que se tratan imágenes y no un patrón de puntos. En la tabla 3.1 se pueden observar las diferencias más notables.

Aspecto	Visión Estéreo	Luz estructurada	Tiempo de vuelo
Complejidad de software	Alta	Media	Baja
Coste de material	Bajo	Alto	Medio
Dispositivo Compacto	Poco compacto	Poco compacto	Muy compacto
Tiempo de respuesta	Medio	Bajo	Alto
Precisión en profundidad	Baja	Alta	Media
Rendimiento con poca luz	Pobre	Bueno	Bueno
Rendimiento con mucha	Bueno	Pobre	Bueno
Consumo de potencia	Bajo	Medio	Escalable
Rango de profundidad	Limitado	Escalable	Escalable

Tabla 2.1. Comparación de técnicas de imagen 3D.

En el mercado también existen otros modelos además del sensor Kinect v2, como son el Asus Xtion [ASUS], el sensor del fabricante PrimeSense en su modelo Carmine [PRIMESENSE] o la primera versión de Kinect [MICROSOFT-KINECT], además de cámaras menos comerciales. Una comparativa rápida entre las características de estos dispositivos nos permite concluir que el sensor Kinect v2 destaca del resto por su mayor resolución de cámaras, su mayor campo de vista, estabilidad de hardware, dar soporte para desarrolladores, y liberar un kit de

desarrollo de software (Software Development Kit, SDK) que permite utilizar una gran cantidad de código que ha sido implementado directamente por Microsoft. Las desventajas son que necesita alimentación directa de la red y una conexión con un pc para su programación mediante USB 3.0 y Windows 8 (como mínimo). Los requisitos para los desarrolladores realmente son algo elevados, pero la liberación del SDK para la libre programación de aplicaciones le da una versatilidad que ninguno de los otros dispositivos posee. Estas han sido realmente las razones por las que se ha elegido este dispositivo para nuestro proyecto. Puesto que, además, está pensado para un uso doméstico, su precio no es excesivamente elevado (150€). Las características técnicas de Kinect for Xbox One se resumen en la tabla 3.2.

Resolución de la cámara RGB (píxeles)	1920 (Horiz.) x 1080 (Vert.)
Resolución de la cámara IR (píxeles)	512 (Horiz.) x 424 (Vert.)
Campo de visión de la cámara IR	70.6 ° (Horiz.) x 60.0 ° (Vert.)
Rango de profundidad	0.5 a 4.5 metros
Software Development Kit:	
	Detección hasta de 6 personas Extracción de la posición de articulaciones Aprendizaje de gestos Detección de caras Kinect Fusion (reconstrucción 3D)

Tabla 2.2. Características de Kinect v2.

2.1.2 OCULUS RIFT DK2

El dispositivo que empleamos para mostrar la información visual pertenece a la compañía Oculus VR, y se trata del modelo Rift Development Kit 2 [OCULUS-2015]. Se trata de la segunda versión de un casco de realidad virtual, que está pensado para su venta a desarrolladores y que se empezó a comercializar a mediados de 2014. Tras el éxito conseguido por la primera versión de Oculus Rift, el salto al mercado por parte de este tipo de tecnologías se está empezando a dar de manera masiva. Una de las principales ventajas de este dispositivo es su precio, que no es excesivamente alto (350$). Se trata de un aparato maduro, con soporte técnico para desarrolladores y con un kit de software (como ocurría con Kinect) que facilita su programación. También hay que añadir que ya no se encuentra a la venta. Durante 2016 se están lanzando al mercado varios modelos de otros fabricantes, como el de Play-Station VR [SONY], HTC Vive [HTC], la iniciativa de

Google Cardboard [GOOGLE] o Microsoft HoloLens [HOLOLENS], aunque éste último modelo realmente se trata de unas gafas de realidad aumentada (posee unos cristales transparentes que dejan pasar la imagen del entorno y se proyecta sobre los mismos un mundo virtual). Algunos de ellos podrían suponer una alternativa al modelo empleado, pero al tener disponibles el sistema de Oculus, se optó por continuar con él. La figura 2.3 muestra el aspecto del dispositivo

Figura 2.3. Oculus Rift DK2

El kit de Oculus que empleamos está formado por dos dispositivos: el casco como tal y una cámara externa (Oculus positional tracker) que debe enfocar en todo momento al casco. La finalidad de utilizar esta cámara es conocer en todo momento la posición espacial de las gafas para traducir el movimiento real de la persona al mundo virtual.

Dicha cámara posee un filtro que permite el paso de la luz en una banda próxima al espectro infrarrojo. Esto le permite leer de manera robusta la información visual emitida por un patrón de 40 leds localizados en la periferia de las gafas (tal y como se puede ver en la figura 2.4) bajo la carcasa de protección (transparente para la luz infrarroja). Para el ojo humano, este proceso es invisible puesto que queda fuera de nuestro espectro visual. La resolución de la cámara es de 752 pixeles horizontales por 480 verticales. El software proporcionado por el SDK de Oculus es capaz de localizar la posición de las gafas con respecto a la cámara gracias a la identificación del patrón de leds. No es necesario que la cámara vea todos a la vez, sino que es suficientemente robusta como para conseguir un buen resultado viendo únicamente unos pocos de ellos. Teóricamente sería necesario un mínimo de 3 puntos para fijar la posición de un objeto en el espacio, pero se desconoce el algoritmo interno que emplea Oculus. Para reconocer que led está viendo la cámara es necesaria una sincronización que se lleva a cabo mediante un pulso de reloj entre la cámara y las gafas por medio de un cable específico. Dotando a cada led de una frecuencia o fase de iluminación distintas se puede identificar cada uno de los leds por separado. Conociendo el mapa de distribución real de los leds en la carcasa y habiendo reconocido cada uno de ellos con la cámara, el sistema es capaz de localizar las gafas en el espacio de manera precisa.

Figura 2.4. Imagen tomada a través de la cámara Oculus positional tracker

Las gafas están formadas, a grandes rasgos, por la pantalla por la que se muestran las imágenes, dos lentes para conseguir un buen enfoque, el hardware de procesamiento, los leds de iluminación, una unidad de medición inercial (inertial measurement unit, IMU) y la carcasa que sujeta todo el conjunto.
La pantalla que utiliza pertenece al fabricante Samsung, y de hecho emplea la misma pantalla que monta el modelo Galaxy Note 3 [IFIXIT-2015]. Se trata de una pantalla de 5.7" Super AMOLED con una resolución de 1920 píxeles horizontales por 1080 verticales (125.77 mm x 70.74 mm) [SPECSMAG-2015]. Por tanto, la geometría del píxel es cuadrada con un tamaño de 0.0655 mm.

Puesto que el ojo se encuentra en una posición muy cercana a la pantalla es necesario utilizar un sistema de lentes que permita un enfoque correcto a la vez que muestre un campo visual elevado para aportar esa sensación de inmersión al usuario al visualizar un entorno tridimensional. Las lentes esféricas tradicionales tienen una curvatura uniforme que causan que la luz que no pasa a través del centro de la lente quede enfocada en distintos puntos a como lo hacen los rayos que sí inciden en el centro. Esto provoca errores de enfoque o aberraciones esféricas y otros tipos de problemas ópticos [FUZOU-2015].

Una forma de solucionar el problema es añadir lentes adicionales para corregir la aberración, pero una forma todavía más eficiente es utilizar otro tipo de lente, de tipo asférico (el que utiliza Oculus). Con ello se reduce al mínimo el número de lentes empleadas (una por ojo), además de ser más delgadas y puede dar lugar a una imagen más nítida. Sin embargo, la lente que emplea Oculus no es suficiente como para corregir totalmente la distorsión, por lo que el SDK aplica una distorsión de tipo barrilete sobre las imágenes que se muestran por pantalla. El conjunto de la lente más el globo ocular corrige dicha distorsión y permite que interpretemos las imágenes de manera visual en nuestro cerebro.

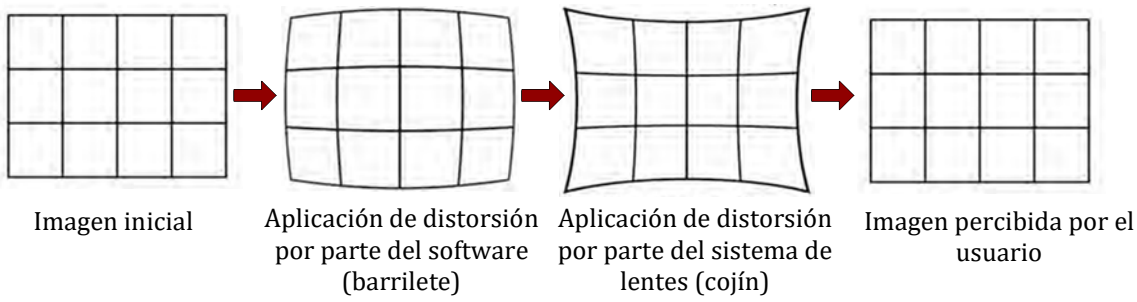


Figura 2.5. Distorsiones aplicadas sobre la imagen inicial para conseguir que el usuario perciba la información sin distorsión

2.2 MONTAJE DEL PROTOTIPO

Uno de los requisitos para poder utilizar el simulador como base de partida para investigaciones futuras en prótesis visuales es que la posición relativa entre la cámara y las gafas debe ser conocida. Las Oculus deben posicionarse delante de los ojos para mostrar la información al usuario, y tienen un tamaño considerable, por lo que se debe buscar una posición para el sensor Kinect que no interfiera con las gafas y que a su vez se encuentre lo más cercano posible a los ojos, puesto que el punto de vista debe ser lo más parecido posible a la visión humana. Con estas premisas se ha colocado la cámara Kinect encima de las gafas. Necesitamos mantener la cámara siempre en la misma posición, puesto que debemos conocer la posición relativa entre ambos dispositivos.

Para fijar ambos dispositivos de manera se emplearon bridas de nylon puesto que es simple y robusto [BADIAS-2016], aunque no se descarta en un futuro diseñar un soporte en plástico.

Figura 2.6. Sistema de sujeción empleado

Para poder utilizar conjuntamente la cámara y las gafas de realidad virtual ha sido necesario calibrar los diferentes dispositivos por separado, así como calibrar la posición y orientación relativa entre ambos una vez están sujetados. Esta tarea fue realizada por Alberto Badías Herbera en su Trabajo Fin de Master (TFM) *Simulación de prótesis visual con sensor RGB-D* [BADIAS-2016]. En el Anexo A se presentan las bases y los resultados finales de dichas calibraciones.

Capítulo

3

DETECCIÓN DE PERSONAS

La detección de personas mediante visión por computador es un tema en auge actualmente. En los últimos años ha habido un gran desarrollo de los sistemas de detección tanto para el esqueleto, como para la cara, gracias a la mejora de las tecnologías de los sensores y la capacidad computacional, lo que ha facilitado la aparición de sistemas robustos y que proporcionan una precisión suficiente tanto del movimiento del cuerpo humano, como de la detección facial, que puede ser de utilidad para un gran número de aplicaciones en ámbitos relacionados con la seguridad o la salud. Uno de esos sistemas es Kinect v2, dispositivo utilizado en este proyecto para realizar la detección de personas, que además de ser suficientemente robusto y preciso, es relativamente económico.

Para realizar la detección del esqueleto, tradicionalmente [WANG-2015] se han utilizado métodos basados en el uso de marcadores de captación de movimiento (Motion Capture System), lo que requiere de la colocación de sensores LED de precisión submilimétrica a lo largo del cuerpo de la persona o personas cuyo movimiento se pretende adquirir u obtener. Esos sensores emiten luz LED que es recogida por las diferentes cámaras infrarrojas posicionadas a lo largo de la zona de captura, para a través de la triangulación, realizar la detección del cuerpo. Sin embargo, gracias a la aparición de dispositivos como Kinect v2, se puede realizar la captación de movimiento a través de una sola cámara RGB-D, sin necesidad de la colocación de sensores sobre la persona. Pese a la mayor precisión de los sistemas basados en marcadores, en muchas aplicaciones la precisión obtenida con la Kinect v2 puede ser más que suficiente: En [WANG-2015] se detalla que los errores están por debajo de 2 mm en el cono central de visión y entre 2 y 4 mm en el intervalo de hasta 3.5 m. El rango máximo de captura permitido por la Kinect v2 es de 4.5 m, donde el error aumenta más allá de 4mm.

A continuación se muestran algunos ejemplos obtenidos de [WANG-2015] donde se compara la detección del esqueleto llevada a cabo con un sistema de captación de movimiento y mediante la Kinect v2 (figura 3.1):

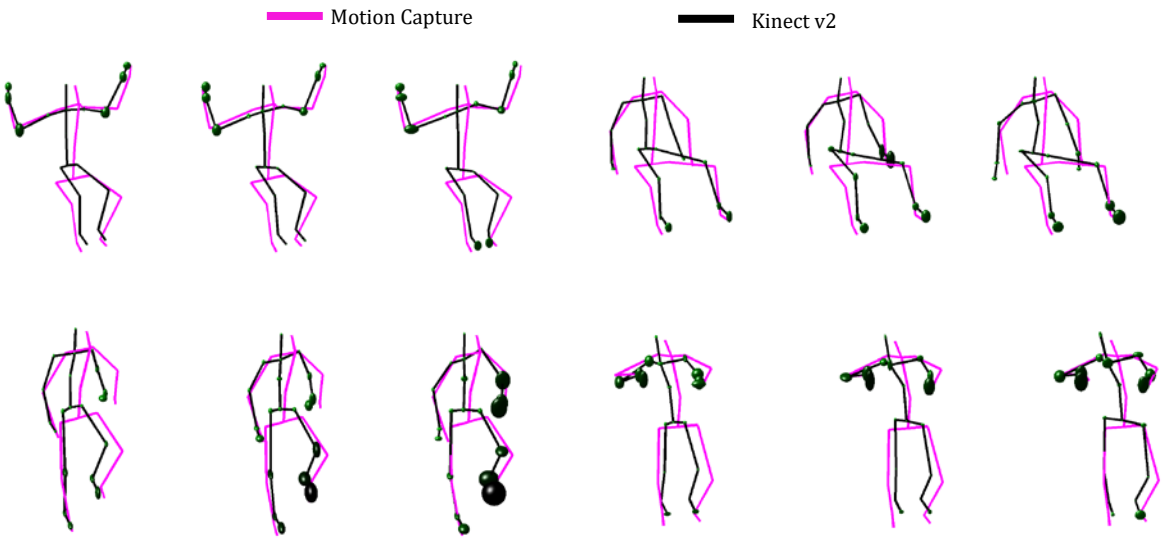


Figura 3.1. Ejemplos de detección realizada mediante sistema Motion Capture y mediante Kinect v2. Reproducidos de [WANG-2015]

La detección de la cara es el primer paso para la realización de algoritmos de análisis facial, que será de utilidad para conocer lo que la gente piensa o cuáles son sus intenciones, toda vez que los ordenadores sean capaces de comprender las expresiones faciales de forma adecuada.

Las dificultades que han existido en los últimos años en la detección de caras, se deben fundamentalmente a grandes variaciones en la escala, la localización, la orientación, la expresión facial, las condiciones lumínicas, etc.

Los primeros trabajos al respecto, señalan que los métodos de detección de caras, pueden clasificarse en 4 grandes grupos [ZHANG-2010]: Los métodos basados en el conocimiento, los basados en características invariantes, los métodos basados en modelos de emparejamiento y los basados en la apariencia. Los primeros utilizan reglas predefinidas, basadas en el conocimiento humano para determinar qué es una cara y qué no lo es dentro de una imagen. El segundo de los métodos trata de encontrar características de la cara que sean invariantes a la posición de ésta y a la iluminación existente. El tercer grupo, utiliza plantillas pre-almacenadas de caras para juzgar si una imagen es una cara o no. Finalmente, el cuarto método "aprende" modelos de caras a través de datos de entrenamiento que constan de imágenes de caras consideradas representativas. Este último método es, generalmente el que mayor rendimiento obtiene, sobre todo gracias al rápido crecimiento de la capacidad computacional y el almacenamiento de datos.
Quizá el mayor progreso del reconocimiento facial en la década pasada se debió al trabajo realizado por Viola y Jones [VIOLA-2004], el cual, gracias a su algoritmo, hizo posible la detección de caras factible en la práctica.

Actualmente, gracias a sensores 3D como la Kinect v2, se puede obtener una solución robusta y suficientemente precisa a la hora de reconocer caras. Pese a que hay otro tipo de sensores 3D más precisos como pueden ser los scanners de alta resolución, por ejemplo Minolta, estos últimos son mucho más caros y tienen una velocidad de adquisición mucho más lenta, lo que limita su uso en muchas aplicaciones. Además en ocasiones, se necesita más de un muestreo para realizar correctamente la detección. En el otro extremo se encuentran sensores 3D como la Kinect v2, que si bien es más ruidosa y tiene menor resolución, permite realizar detecciones en tiempo real y a un precio mucho más competitivo. Según [LI-2013] utilizando los datos de profundidad y de color proporcionados por Kinect v2 se pueden reconocer caras en diferentes posiciones, con distintas expresiones y bajo cambios de iluminación utilizando algoritmos compactos y escalables con unos ratios de reconocimiento de casi el 97%. Estos datos justifican el uso de este tipo de sensores 3D de baja resolución para la detección facial de forma robusta en nuestro caso.

3.1 DETECCIÓN DE CARAS

El primer paso para poder realizar la detección de caras ha sido realizar una búsqueda de las herramientas software disponibles para tal efecto. Tras la búsqueda de resultados recientes de la investigación en visión por computador, nos hemos decantado por las herramientas proporcionadas por Microsoft en el Software Development Kit (SDK) de Kinect 2.0 para Windows [KINECT]. Analizando las diferentes herramientas y funciones disponibles en la SDK, se ha optado por utilizar las librerías Face-Basics. La implementación de la detección se ha basado en una modificación del código de ejemplo Face-Basics D2D incluido con el SDK.

El siguiente paso ha sido la comprensión detallada de este código de ejemplo con el fin de poder realizar las modificaciones necesarias para adecuarlo a este proyecto. A continuación, se van a mostrar de forma resumida los principales datos que nos proporciona *Face-Basics D2D*, que serán los que habrá que procesar posteriormente para realizar la representación mediante fosfenos.

El software de detección disponible en la SDK procesa la imagen RGB-D y muestra por pantalla en tiempo real la posición de la cara mediante un cuadrado superpuesto a la imagen que está capturando Kinect. De esta forma, mediante los datos que proporciona de las 4 esquinas de ese cuadrado, ya se puede conocer la posición de la cara en todo momento. Buscando en la representación interna del resultado de procesamiento hemos visto que se proporcionan además, datos de la orientación 3D de la cabeza respecto a la cámara. Además el sistema reconoce si la

persona está sonriendo o no, si lleva o no gafas, si la boca está abierta o no, si el ojo izquierdo o el derecho están cerrados, etc.

A la vista de la información obtenida de la cara se ha analizado que información resultaría más útil a una persona ciega para establecer comunicación con su interlocutor, con la premisa de que sea una información procesable de forma icónica en una prótesis biónica de visión. Por un lado se ha pensado que analizando la orientación de la cara con respecto al observador se puede inferir al menos de forma aproximada si el interlocutor está mirando directamente. Por otro lado la detección de sonrisa también parece una información relevante para un posible usuario de la prótesis. Por el momento, se ha decidido que aparte de la detección de la cara, la detección de sonrisa y la orientación 3D de la mirada son la información adicional a codificar.

Por tanto, los datos de la detección de cara que serán procesados más adelante serán: la posición de la cara (con la que sabremos donde se encuentra ésta en todo momento), la posición de los ojos, la posición de la boca (concretamente de los extremos izquierdo y derecho de la misma), la orientación de la cabeza (con la que podremos saber si la persona detectada nos mira o no), y si esa persona nos sonríe o no. Toda esta información resulta de gran ayuda para una persona ciega a la hora de poder interactuar con alguien de su entorno.

A la hora de realizar la detección, se optó por trabajar sobre una captura de vídeo, en lugar de realizar la representación en línea de lo que se estaba detectando. De esta forma, se evitaba que la representación se viera afectada por posibles errores en la detección, y se trabajaba en un entorno mucho más controlado en el que se sabía en todo momento que lo que se estaba representando era acorde a lo que se estaba detectando. Para poder realizar esto, se modificó el software de detección para almacenar todos los datos que nos interesaban en un fichero, que posteriormente se lee para realizar la representación icónica.

La solución adoptada ha sido generar un fichero para cada frame del vídeo que incluye toda la información necesaria para la representación en fosfenos: información semántica de la cara y posición de cara, ojos y boca. Así pues, a la hora de representar, basta con recorrer todos los frames e ir obteniendo los datos necesarios para poder procesarlos posteriormente.

El siguiente paso después del almacenamiento y lectura de los datos de detección ha sido realizar un primer procesamiento de los mismos. Para comprobar que tanto la detección como el procesamiento eran correctos, se optó por realizar una pequeña representación a modo de testeo, de forma superpuesta a la imagen original.

Para representar la cara, se utilizó la función *circle* disponible en OpenCV. Esta función permite representar una circunferencia a partir de su centro y su radio. Para poder pasarle como parámetro el centro de la cara, hubo que calcularlo a partir de los datos de las 4 esquinas del cuadrado que simulaba la posición de la cara, obtenidos en la detección.

En cuanto a los ojos, se utilizó la misma función *circle* solo que en este caso, el radio es mucho más pequeño, y el centro de los ojos es la propia posición de los mismos obtenidos de la detección.

A la hora de representar la boca, se tendrá en cuenta si la persona detectada está sonriendo o no. Por tanto, es necesario conocer si la propiedad "contento" es cierta o no. En el caso de que lo sea se representará una sonrisa utilizando la función *ellipse* de OpenCV, en la cual partiendo del centro y fijando radio mayor, radio menor y ángulo girado se puede realizar una elipse (en nuestro caso es una semielipse). Ese centro hubo que calcularlo previamente como el punto medio entre los dos extremos de la boca, que son los dos puntos que proporciona la detección. En el caso de que la persona no esté sonriendo se representará una línea recta teniendo como punto de inicio el extremo izquierdo de la boca y como final el extremo derecho de la misma, los dos parámetros de entrada de la función *line* de OpenCV con los que se representó la boca en este caso.

A continuación se va a exponer como se ha decidido representar el hecho de que la persona detectada esté mirando al usuario o no, para lo que se propone la siguiente idea: A través de la orientación de la cabeza, se conocerá si la persona está mirando o no, por lo que en el caso de que esto suceda, se representarán los ojos y la boca. Mientras que si la persona no está mirando, no se representarán ni ojos ni boca.

Para llevar a cabo esta idea es necesario conocer qué datos de orientación proporciona el sistema de detección, y no son otros que los cuaternios. Los cuaternios son una representación algo abstracta pero muy potente que se basa en que cualquier giro (o sucesión de giros) 3D puede ser representado a partir de un eje y un ángulo. En este caso, el dato que proporciona la detección es un vector de 4 elementos en el que los 3 primeros definen la dirección del eje (componentes vectoriales) y el último elemento define el ángulo de giro (componente escalar). Esto se muestra en la ecuación 3.1

$$Q = [q_0, q_1, q_2, q_3] = [k \sin\frac{\theta}{2}, \cos\frac{\theta}{2}] \qquad \text{(ecuación 3.1)}$$

Donde k es un vector unitario que define el eje de giro en 3D y θ es el ángulo girado.

La orientación de la cara viene expresada en el sistema de referencia de la cámara, donde un giro nulo corresponde con una cara fronto-paralela al plano imagen. Si asumimos que observamos personas en posición normal (que no están boca abajo por ejemplo) es suficiente conocer el giro relativo entre el eje óptico de la cámara y la dirección 3D de la cara para estimar si la persona está mirando a la cámara.

Por lo tanto, de todo ese vector, la componente que nos interesa es la componente escalar. Así pues, basta con calcular el arcos de la cuarta componente del vector, y el resultado multiplicarlo por dos, para obtener el ángulo entre la cara y el plano imagen. El siguiente paso es definir con ese ángulo el rango dentro del cual se considera que la persona está mirando. Tras realizar la experimentación necesaria, se decidió que entre -25 y 25 grados la persona "miraba", por lo que tanto ojos como boca se representan. Fuera de ese rango, se considera que la persona no mira y por tanto, tanto ojos como boca no se representan.

En la figura 3.2 se van a mostrar algunos ejemplos en los que se puede ver más claramente los aspectos comentados anteriormente:

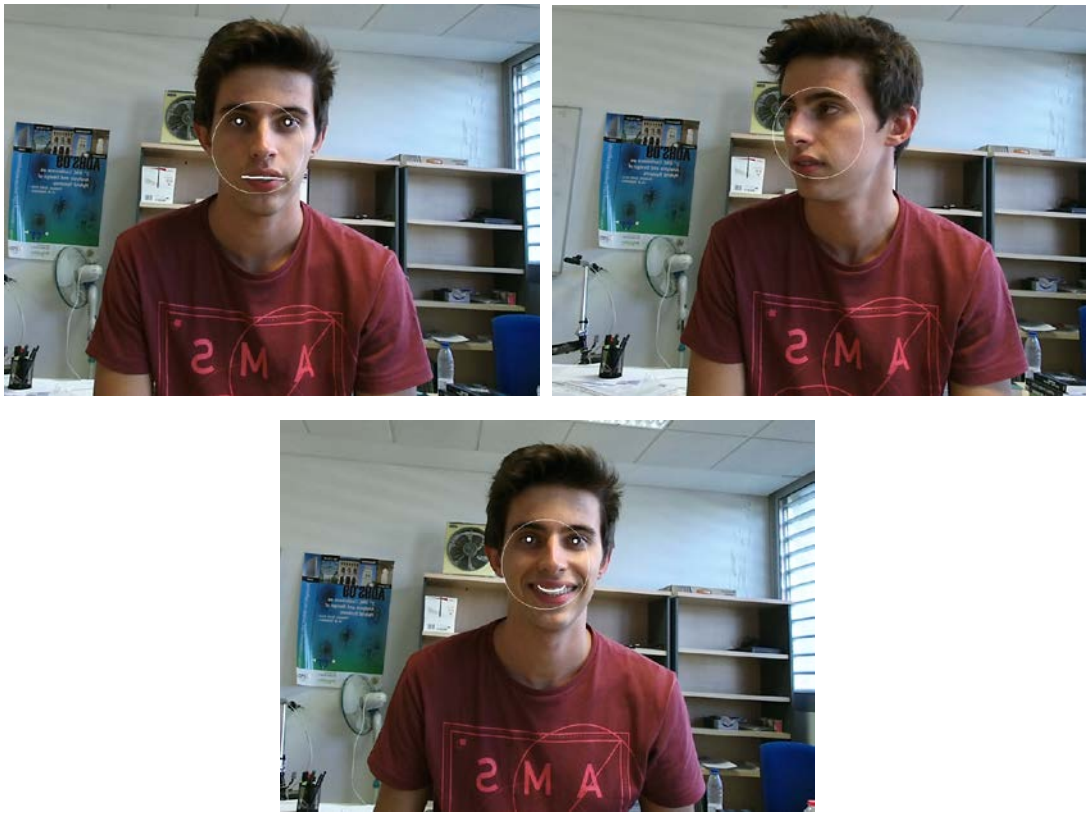


Figura 3.2. Ejemplos de detección de la cara

En la primera imagen se observa como el grado de rotación de la cabeza está dentro de los límites especificados, por lo que "nos está mirando", es por ello que tanto los ojos, como la boca aparecen pintados en la representación.

En la segunda imagen, podemos ver como la orientación de la cabeza se ha salido fuera del rango que se había establecido previamente, por lo que ahora "no nos estaría mirando", así pues, los ojos y la boca no estarían representados.

Finalmente, en la última imagen vemos como al estar sonriendo, la representación de la boca ha cambiado, pasando de ser una línea recta, a ser una semielipse que simula una sonrisa.

Al ver las imágenes se ha corroborado que tanto la detección, como el posterior procesamiento de los datos funciona correctamente, por lo que el siguiente paso en cuanto a la cara es realizar la representación icónica mediante fosfenos, la cual se abordará en los siguientes capítulos.

3.2 DETECCIÓN DEL CUERPO

El siguiente aspecto a tratar va a ser la detección del cuerpo. Para poder realizarla, se ha modificado el código de ejemplo Face-Basics D2D que también se ha utilizado en la detección de caras. En la fase de aprendizaje se observó que dado que la detección de caras de Kinect utiliza internamente la detección del cuerpo se podía re-utilizar el mismo software.

El procedimiento ha sido similar al usado para la detección de caras, es decir, se guarda la información relativa a la detección del esqueleto en un fichero, con el fin de trabajar en un entorno mucho más controlado. Posteriormente se lee ese fichero y se realiza un primer procesamiento de los datos obtenidos. A continuación se expone el contenido de estos datos:

En primer lugar se tiene un vector de datos que supone la posición X e Y de cada una de los 25 puntos (joints) del esqueleto que proporciona el sistema de detección, la mayoría de esos puntos se corresponden con las distintas articulaciones del cuerpo (codo, hombro, muñeca, rodilla, tobillo…) pero también se dispone de otro tipo de puntos como son por ejemplo la posición de la punta del dedo corazón o la posición de la punta del dedo pulgar. Otro vector de datos corresponde a las distintas profundidades de esos 25 puntos representativos del esqueleto, y por último un vector que indica si esos joints han sido detectados y seguidos en la secuencia de video o no.

Así pues, una vez leídos los datos, es necesario procesarlos. Para saber si tanto la lectura y detección de los datos como su procesamiento eran correctos, se optó por realizar una pequeña representación a modo de testeo, superpuesta a la imagen original (de forma similar a lo realizado para las caras). Esa representación se realizó utilizando la función *line* disponible en OpenCV. Los parámetros que toma

esta función son, básicamente, la imagen donde se va a realizar la representación, el punto inicial y final de la línea que se dibuja. Esos puntos inicial y final serán los distintos joints que han sido obtenidos en la detección. Así pues, solo representaríamos mediante la función *line* si los dos joints implicados en cada "eslabón" estaban detectados y seguidos correctamente.

En la figura 3.3 se muestran algunas imágenes que ilustran la aplicación anterior:

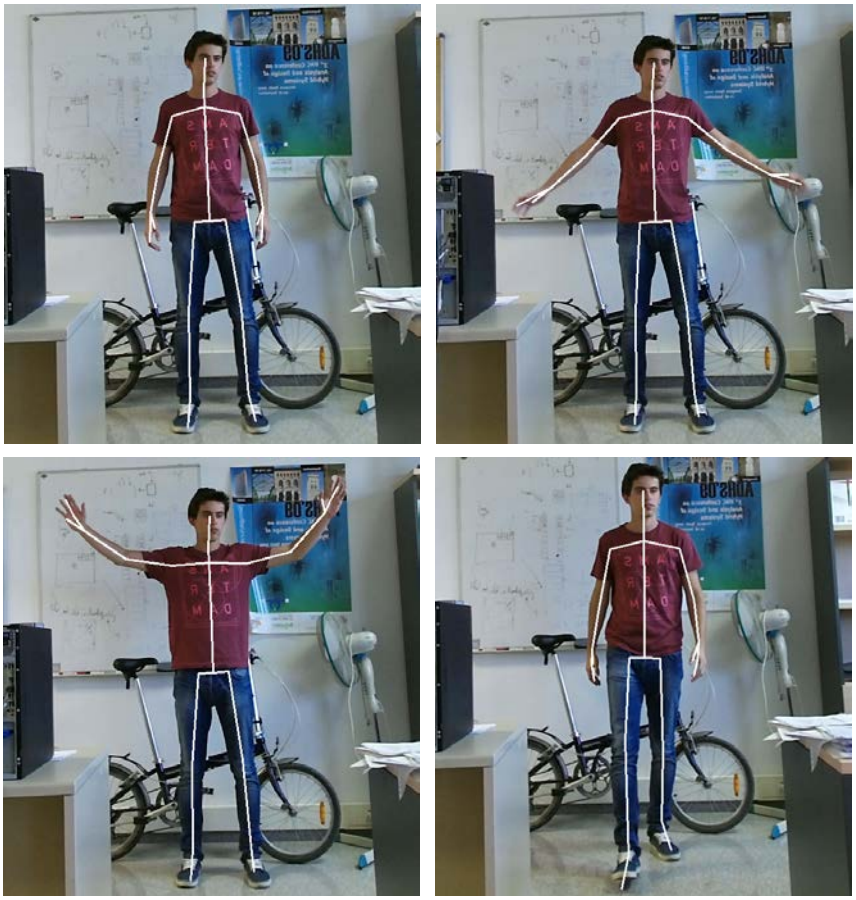


Figura 3.3. Ejemplos de detección del esqueleto. A la imagen se superpone la representación con líneas que unen cada uno de las articulaciones y puntos extremos del modelo de detección

En estas imágenes, se puede ver cómo tanto la detección como el procesamiento son correctos, por tanto, ahora el siguiente paso es la representación icónica del cuerpo y la cara mediante fosfenos.

Capítulo

4

REPRESENTACIÓN ICÓNICA MEDIANTE FOSFENOS

Antes de abordar la implementación correspondiente a la representación mediante fosfenos, se va a realizar una breve explicación de las distintas prótesis visuales disponibles en la actualidad y se va a mostrar también los mapas de fosfenos que se han utilizado a la hora de realizar la simulación de esas prótesis en este proyecto.

4.1 MAPA DE FOSFENOS

Actualmente hay 2 modelos de prótesis aprobados por la Unión Europea para el tratamiento de retinitis pigmentosa [WEILAND-2014]. Uno de ellos (Argus II) pertenece a la empresa estadounidense Second Sight Medical Products, Inc. Se trata de un implante epirretinal de 60 electrodos con un ángulo de visión máximo de 20° en diagonal, que queda posicionado con respecto al ángulo visual como se observa en la figura 4.1. La separación entre los centros de los electrodos es de 525 μm (tanto en horizontal como en vertical), que corresponde a unos 1.88° de campo visual, suponiendo que 1° equivale a 280 μm de longitud en la retina [HIRSH-1989]. El segundo sistema aprobado en Europa es el Alpha-IMS de los alemanes Retina Implant AG. Se trata de un implante subrretinal de 1500 fotodiodos que capturan la luz proyectada sobre la retina y la transforman en señales eléctricas. En este caso no hay tratamiento de la imagen capturada, pero nos sirve para estimar la resolución que puede alcanzarse con este tipo de prótesis. El implante tiene un tamaño de 3 x 3 mm y los electrodos tienen un diámetro de 50 μm [AG-2015], con una distancia entre centros de electrodos de 70 μm que equivale a una resolución angular de 0.25°. Así pues, para poder tratar la imagen (como es el caso) deberíamos decantarnos por un sistema similar al de Argus II, pero con mayor resolución.

Los ejemplos anteriores nos revelan que las prótesis que se comercializan hoy en día todavía trabajan con campos visuales muy reducidos (figura 4.1), aunque la densidad de electrodos puede ser bastante elevada (así lo demuestra el implante Alpha-IMS).

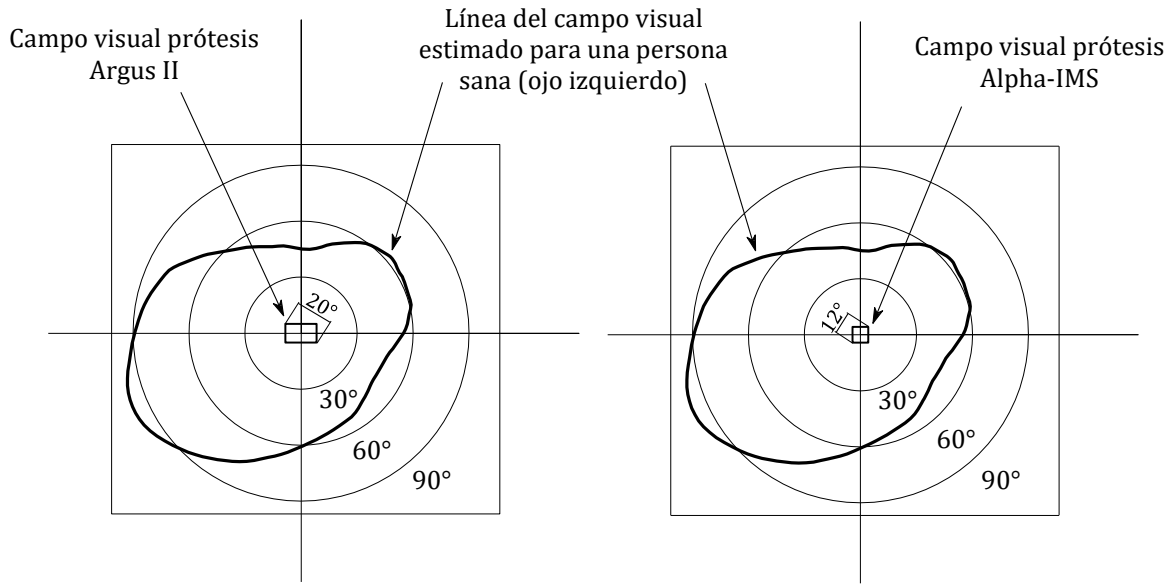
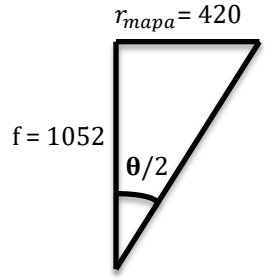


Figura 4.1. Comparación entre el campo visual permitido por dos modelos de prótesis comerciales con respecto al campo visual estándar de una persona sana para el ojo izquierdo

Con vista a desarrollos futuros, y puesto que estamos en fase de investigación, no tendría ningún sentido trabajar con las características de los dispositivos actuales, por lo que para nuestro simulador se ha utilizado un campo visual circular de 43°. El cálculo del mismo se ha llevado a cabo gracias a los datos de calibración de Kinect (anexo A).

A raíz de esas calibraciones, obtenemos una distancia focal (f) de la cámara RGB de 1052 píxeles. Así pues, teniendo en cuenta que el mapa de fosfenos tiene un radio de 420 píxeles, el campo de vista del simulador empleado se calcula (figura 4.2):



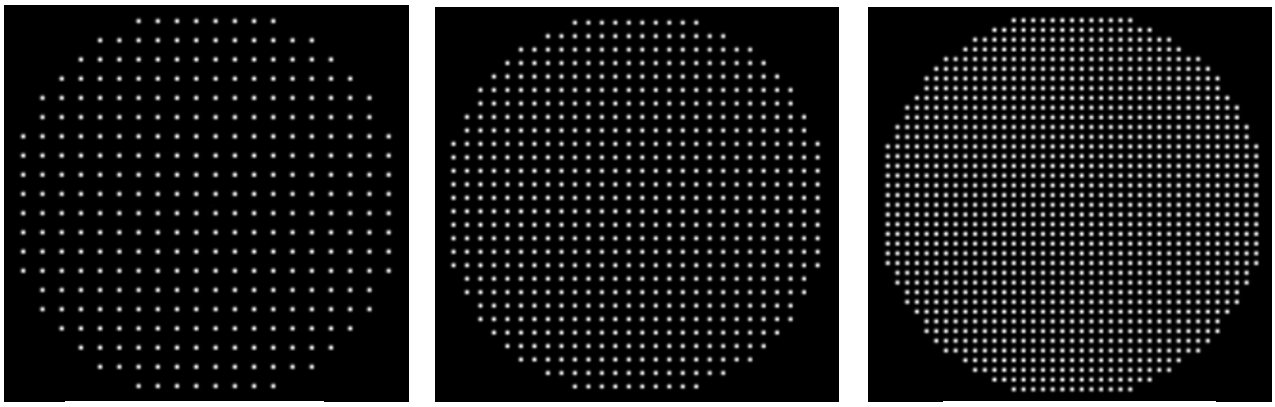
$$\theta = 2\tan^{-1}\frac{r_{mapa}}{f} = 2\tan^{-1}\frac{420}{1052} = 43°$$

Figura 4.2. Cálculo del campo de vista del mapa de fosfenos utilizado

Una vez visto el campo visual del simulador utilizado, se van a mostrar los diferentes mapas de fosfenos utilizados a la hora de realizar las representaciones fosfénicas, con tres resoluciones distintas, ya que es este el aspecto más

importante toda vez que se disponen de prótesis que permiten realizar el tratamiento de la imagen. Estos mapas se muestran en la figura 4.3:

Baja: Distancia entre centros (pixeles): 40; Radio de iluminación (pixeles): 14
Media: Distancia entre centros (pixeles): 29; Radio de iluminación (pixeles): 14
Alta: Distancia entre centros (pixeles): 21; Radio de iluminación (pixeles): 14



Resolución baja Resolución media Resolución alta

Figura 4.3. Distribución de los fosfenos en cada uno de los mapas

4.2 ESTRATEGIA GENERAL DE REPRESENTACIÓN

Una vez diseñadas las distintas opciones en cuanto a mapas de fosfenos, se va a exponer la estrategia general que se ha llevado a cabo para realizar la representación icónica mediante fosfenos.

A la hora de crear tanto el mapa de fosfenos, como la forma de estos, se ha utilizado un diseño modular, buscando mayor facilidad a la hora de modificar los distintos aspectos como por ejemplo la resolución del mapa de fosfenos, o la forma del fosfeno en cuestión.

Así pues, el primer paso ha sido implementar una función que permita crear fosfenos. Se decidió que los fosfenos fueran circulares y la iluminación de los mismos siguiera una distribución gaussiana de tal forma que en el centro el brillo sería más intenso que en la parte exterior. Se ha creído adecuado utilizar este tipo de representación ya que es la que mejor refleja lo que vería una persona ciega con una prótesis visual. La función utilizada para este fin es *genPhoshenSprite* la cual permite cambiar tanto el radio de iluminación del fosfeno como el color del mismo.

El siguiente paso es generar el mapa de fosfenos. Esto se ha hecho mediante la función *getPhosphenesPos* la cual almacenará la posición de cada fosfeno en una matriz. Los parámetros de entrada de esta función son: el radio exterior del mapa

de fosfenos, la separación entre fosfenos en dirección X e Y, y el centro, $(x_0,\ y_0)$, a partir del cual se comienza a construir el mapa.

Una vez se ha realizado el mapa, aparece la siguiente problemática: Cada punto que se quiera representar mediante fosfenos, debe corresponder con uno de los fosfenos del mapa. Es por ello que se ha optado por crear una look-up table en la cual conociendo las coordenadas del pixel que se quiere representar, se obtenga el fosfeno más cercano a ese punto. Todo esto se almacenará en una matriz. De esta forma, además de obtener una correspondencia pixel-fosfeno, se consigue una mayor velocidad de ejecución, ya que esta matriz se creará una vez al comienzo de la ejecución, y después bastará con consultarla cada vez que se quiera representar, para saber que fosfeno se debe encender. La función realizada para conocer el fosfeno correspondiente a cada pixel es *getxyPhosphene* en la cual simplemente hay que introducir el punto que se desea conocer y la look-up table y la función devolverá el fosfeno en cuestión.

Por último, hace falta una matriz en la que se tengan almacenados todos los fosfenos y en la que, en función de lo que se necesite, se seleccione si encender o no cada fosfeno. Esa matriz se llama *MatPhosphene*. Así pues, una vez que cada fosfeno tenga asignado un "encendido" o un "apagado", bastará con recorrer esa matriz y encender o apagar cada fosfeno en función de lo que se indica para cada uno de ellos.

Llegados a este punto, ya se ha explicado la estrategia general que se sigue para afrontar la representación. De aquí al final del capítulo se va a exponer la implementación específica que se ha realizado para representar tanto la cara como el esqueleto.

4.3 REPRESENTACIÓN DE LA CARA

En cuanto a la representación de la cara se han propuesto tres tipos de representación posibles:

1. Representar ojos, boca, y cara.
2. Representar ojos, boca y cejas.
3. Representar solo ojos y boca.

Para evaluar cuál de los tres tipos de representación se utilizará, se van a mostrar diferentes ejemplos de cada una, con el fin de evaluar las posibles ventajas e inconvenientes de las mismas, y poder así, elegir la mejor opción de cara a una correcta interpretación en el mapa de fosfenos.

Debemos partir de la base de que la representación de cualquier parte del cuerpo debe aportar, en primer lugar, algún tipo de información útil frente a no representarla, ya que en caso contrario carecería de sentido utilizar los escasos recursos a nivel de fosfenos disponibles en la práctica. Además no debe confundirse con otras partes de la representación, lo que podría llevar a un error de interpretación por parte del usuario.

La representación de los ojos se hace simplemente encendiendo los dos fosfenos correspondientes. Por su parte, la boca se representa mediante una "línea recta" que se ha implementado por medio del encendido de varios fosfenos, si la persona no sonríe, o mediante una "semicircunferencia" simulada por el encendido de varios fosfenos, en el caso de que esta si sonría. Tanto ojos como boca, se representarán de este mismo modo en todos los casos que se exponen a continuación.

Dicho esto se va a evaluar cada tipo de representación por separado:

Tipo 1: En este primer caso, se ha optado por representar la cara mediante una "circunferencia". A priori esta podría ser la que parece más intuitiva, ya que nos permite diferenciar claramente la cara de otras partes del cuerpo. Uno de los inconvenientes es que en la representación por fosfenos, se tiene una representación discreta (debido a la disposición fija de las posiciones de estos), por tanto, cada punto que se quiere representar hay que aproximarlo al fosfeno más cercano. Esto da lugar a que, en algunas ocasiones, obtengamos algo que difiere mucho de ser una circunferencia, sobre todo, a resoluciones más bajas.

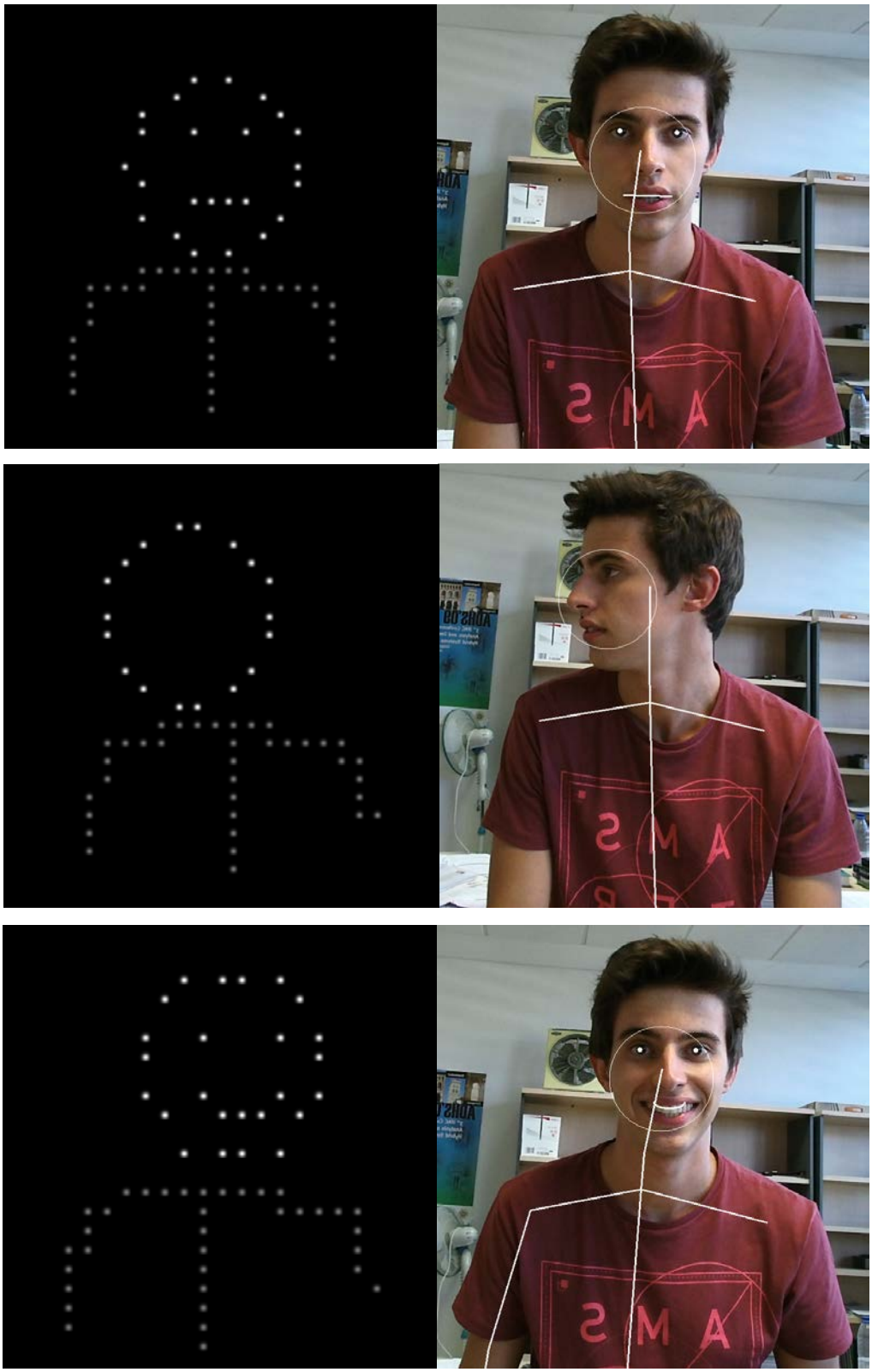


Figura 4.4. Ejemplos de representación para el tipo 1 mediante fosfenos de la cara detectada en la imagen de la derecha.

En el primer caso de la figura 4.4 vemos como la persona no sonríe por lo que la boca es una "línea recta", además como está dentro del rango en el que se considera que está mirando, tanto la boca como los ojos se muestran en la representación icónica mediante fosfenos.

En el segundo caso, se puede ver como la persona no está mirando por lo que tanto boca como ojos no se representan, sin embargo, sí se representa la cara, lo que permite saber hacia qué lado mira la persona detectada. Esto proporcionará más información al usuario sobre qué ocurre en su entorno.

En el último caso, se puede observar que la persona sonríe, es por ello, que la boca pasa a ser una "semicircunferencia", simulando una sonrisa

Tipo 2: Otra opción de representación se basa en representar únicamente las cejas encima de los ojos, además de la boca, lógicamente. Este tipo de representación, podría aportar al usuario la suficiente información de donde se encuentra la cara, evitando al mismo tiempo, las posibles confusiones que se pueden dar en el tipo 1 entre la cara y otras partes del cuerpo. A continuación se muestran algunos ejemplos (figura 4.5)

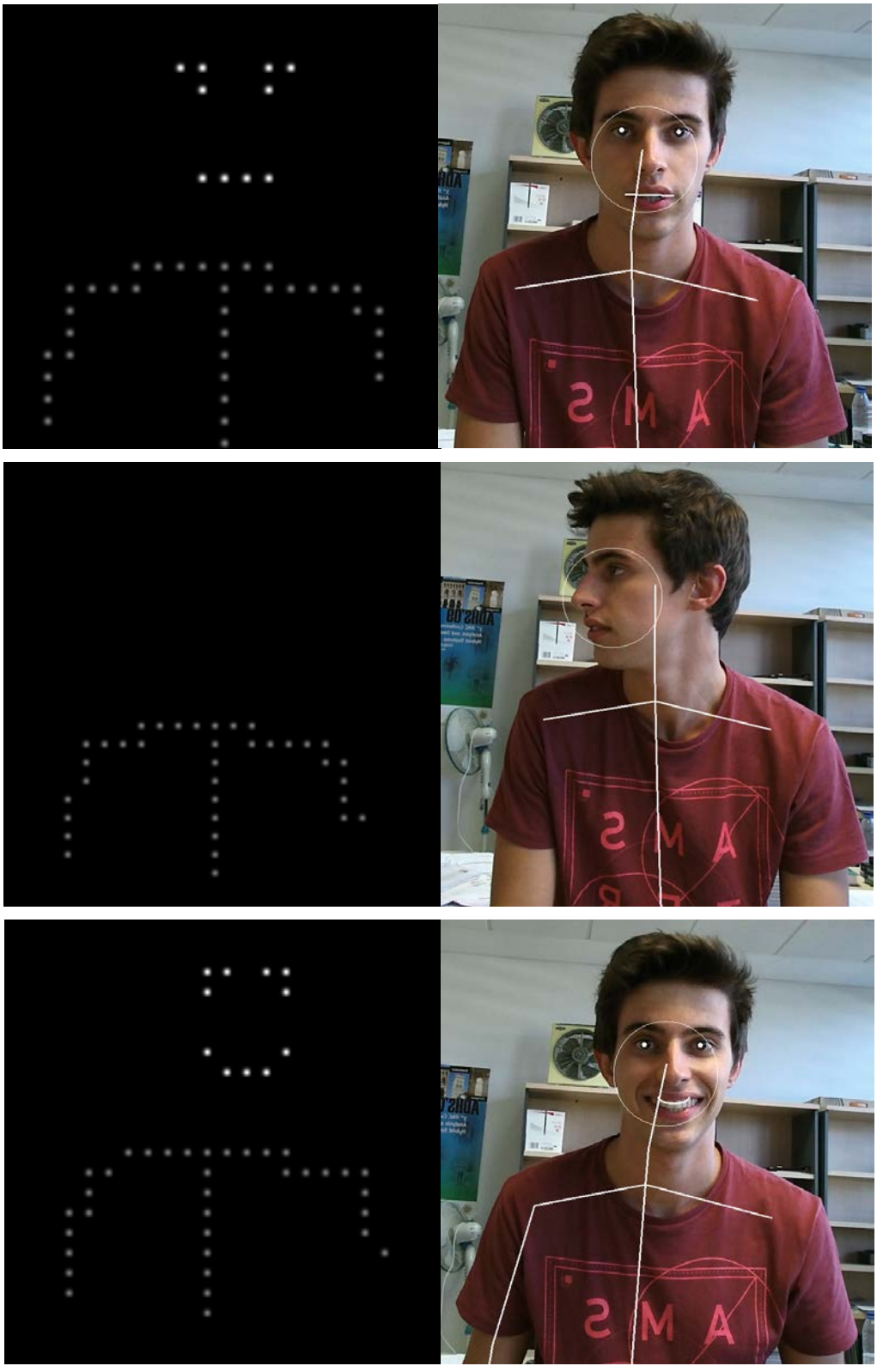


Figura 4.5. Ejemplos de representación para el tipo 2 mediante fosfenos de la cara detectada en la imagen de la derecha.

En la primera imagen de la figura 4.5 se aprecia como la persona está mirando, pero no sonríe

En el segundo caso de la figura 4.5 y de forma análoga a lo que ocurre en el tipo 1, cuando la persona no mira al usuario de la prótesis, no se dibuja ni ojos ni boca. Sin embargo aquí, al no poderse detectar las cejas, la posición de la cara no se puede conocer si la persona no mira hacia el usuario.

En el último caso la persona está sonriendo, por lo que la representación de la boca cambia consecuentemente.

Tipo 3: Finalmente, se propone el tercer tipo de representación, en el que solo se representan ojos y boca. Aunque esta representación puede ser algo pobre en ocasiones, la información que proporciona es la necesaria, ya que son los ojos y la boca los que realmente aportan la información imprescindible al usuario de la prótesis. Posiblemente este tipo de representación es la más adecuada cuando la resolución es baja. (Figura 4.6)

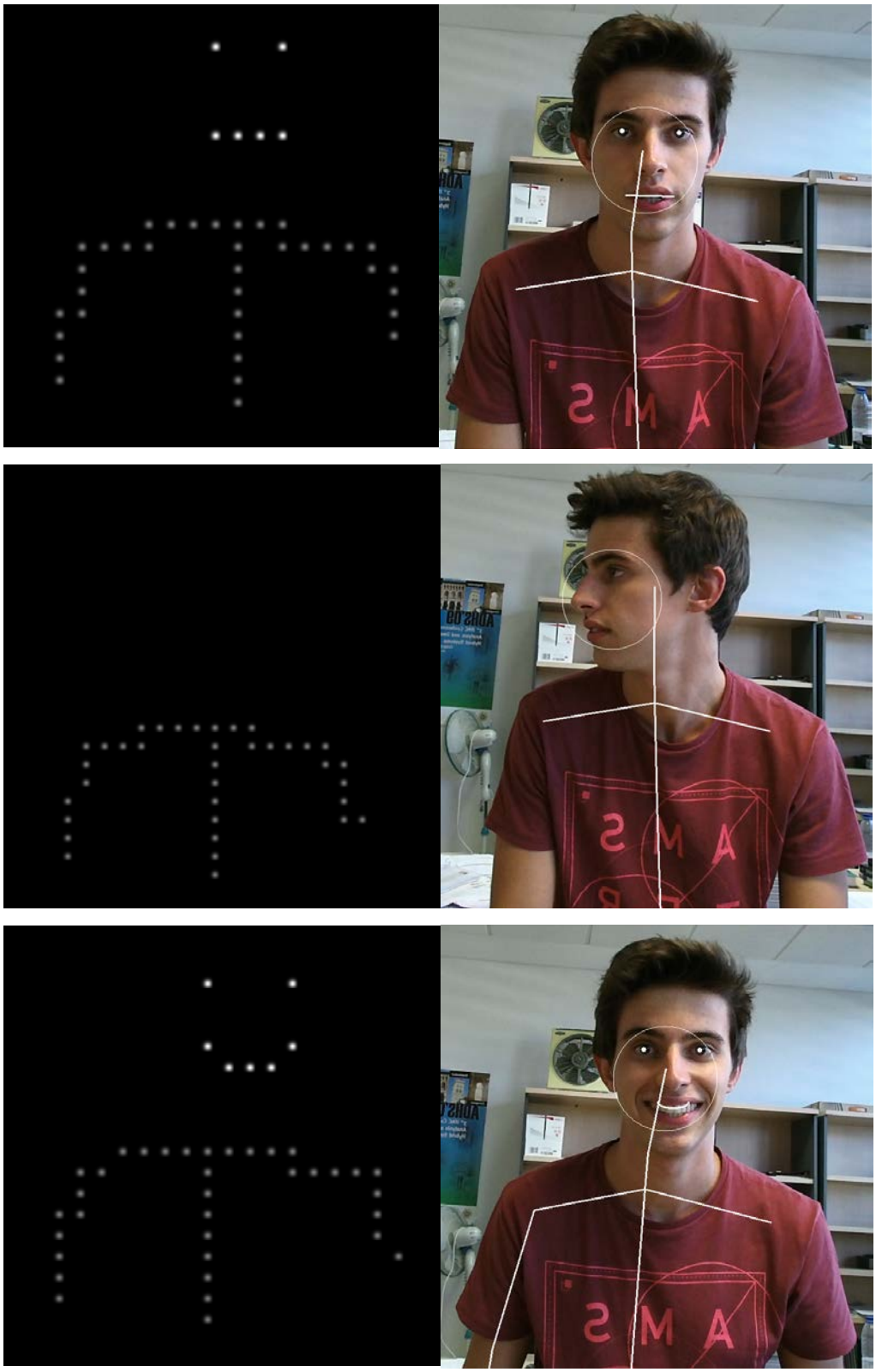


Figura 4.6. Ejemplos de representación para el tipo 3 mediante fosfenos de la cara detectada en la imagen de la derecha.

En el primer caso de la figura 4.6 vemos como la persona está mirando pero no sonríe, por lo que se representan tanto ojos como boca, y esta última será una "línea recta"

En el segundo caso no se representa ni boca ni ojos, ya que la persona no está mirando hacia el usuario.

En el último ejemplo, la persona sonríe, por lo que la representación de la boca ha variado, y ahora simula una sonrisa.

Una vez vistos los distintos tipos de representación propuestos se va a explicar cómo se ha realizado la representación mediante fosfenos de las distintas partes que componen la cara, es decir, los ojos, la boca (tanto si sonríe como si no), el contorno de la cara (tipo 1) y las cejas (tipo 2):

En cuanto a los ojos simplemente es necesario conocer el fosfeno correspondiente a cada ojo, sabiendo la posición de estos, y encenderlo.

Para representar la boca hay dos opciones, en función de si la persona sonríe o no. Si esta sonríe habrá que representar la semicircunferencia sabiendo que cualquier punto (x, y) de la misma se puede representar como (ecuación 4.1):

$$\begin{aligned} x &= x_{ini} + r * \cos\theta \\ y &= y_{ini} + r * \sin\theta \end{aligned} \qquad \text{(ecuación 4.1)}$$

Siendo r, el radio y (x_{ini}, y_{ini}) el centro de esa circunferencia, que en este caso particular será el punto medio de la boca (calculado a partir de los extremos de esta que se han obtenido en la detección). Así pues, basta con ir variando el ángulo θ un delta constante entre 0 y π radianes, y calcular el fosfeno correspondiente a cada punto (x, y) para poder representarlo.

En el caso de que la persona no sonría, la boca se representará mediante una recta basándose en el hecho de que cualquier punto (x, y) de la misma se puede representar como (ecuación 4.2):

$$\begin{aligned} x &= x_{ini} + \left(x_{fin} - x_{ini}\right) * \lambda \\ y &= y_{ini} + \left(y_{fin} - y_{ini}\right) * \lambda \end{aligned} \qquad \text{(ecuación 4.2)}$$

Siendo λ el "paso" y $(x_{fin} - x_{ini})$ el vector director de la recta. Por tanto, lo que hay que hacer es ir variando λ un delta constante entre 0 y 1, y calcular el fosfeno correspondiente a cada punto (x, y) para poder representarlo.

En cuanto a la representación de la cara para el primer tipo de representación, hay que realizar el mismo procedimiento que se ha seguido para la boca sonriente, pero en esta ocasión en lugar de variar θ entre 0 y π radianes, se variará entre 0 y 2π radianes.

Por último, a la hora de representar las cejas, se ha utilizado el mismo procedimiento que para la boca no sonriente, pero referenciando el punto inicial y final de la recta, a la posición de los ojos, añadiendo un desplazamiento vertical fijo sobre estos.

4.4 REPRESENTACIÓN DEL CUERPO

La representación del esqueleto, se ha realizado utilizando segmentos de recta y siguiendo la misma estrategia que se ha explicado en 4.3. Sin embargo, en este caso, se propone una novedad en cuanto a la representación: utilizar la intensidad de los fosfenos como medio para transmitir al usuario la idea de profundidad o de distancia ante una posible interacción.

Como ya se ha comentado anteriormente, el sistema de detección que se ha utilizado, proporciona datos de profundidad para cada *joint*. Por tanto no solo tenemos la proyección en la imagen de la articulación o punto del cuerpo, sino también su distancia al observador. Se va a utilizar esto para definir tres niveles de intensidad en función de lo que representa para el usuario de la prótesis una distancia mayor o menor.

En el primer nivel, caso en el que la persona se encuentre alejada del usuario, la intensidad será baja (primera imagen de la figura 4.7). Esa baja intensidad representará que hay una persona en las proximidades del usuario de la prótesis, pero que no tiene por qué interactuar con este. Sin embargo, el usuario es consciente que hay alguien, por lo que podría ser este quien comenzara una interacción.

En el segundo caso, en la que la intensidad aumenta, representa el hecho de que la persona detectada está a una distancia que se considera “normal” para mantener una conversación con el usuario de la prótesis (segunda imagen de la figura 4.7).

Finalmente, la distancia más cercana, representa que la persona detectada, está a una distancia que se considera “normal” para saludar dando la mano (tercera imagen de la figura 4.7). En este caso los fosfenos del cuerpo alcanzan la intensidad máxima, mientras que la mano permanece en un nivel de intensidad menor, para que el usuario de la prótesis sea capaz de identificarla sin dificultad y pueda saludar a su interlocutor en el caso de que éste le ofrezca su mano para tal fin.

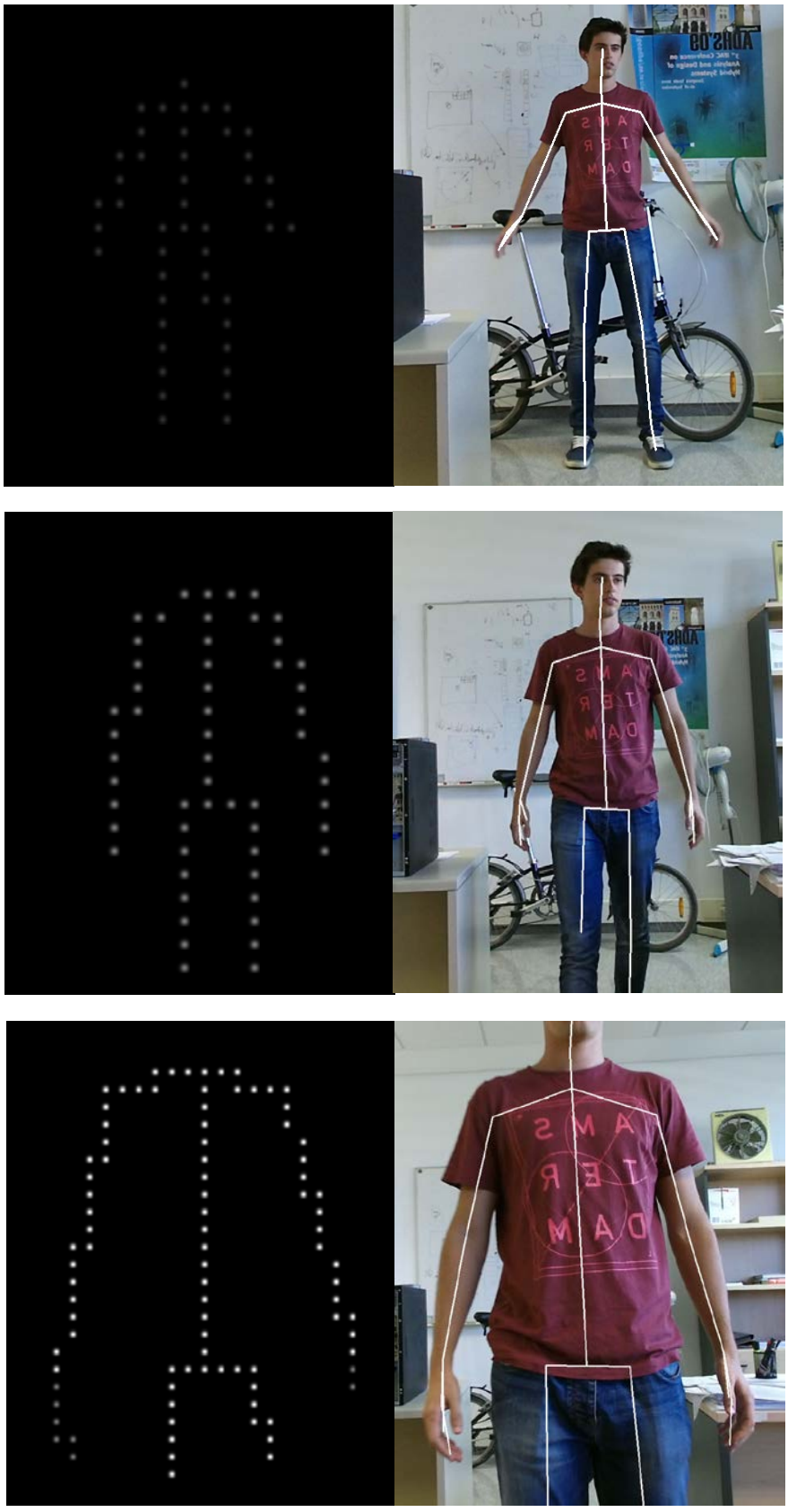


Figura 4.7. Ejemplos de representación mediante fosfenos del esqueleto detectado en la imagen de la derecha

En las diferentes representaciones mediante fosfenos mostradas anteriormente, tanto en este apartado como en el anterior sobre la representación de la cara se ha empleado una resolución media. En el siguiente capítulo se van a mostrar más ejemplos de estas representaciones para las diferentes resoluciones expuestas anteriormente en este capítulo.

4.5 INTEGRACIÓN EN UN SISTEMA DE REALIDAD VIRTUAL

Una vez se ha conseguido realizar la representación icónica mediante fosfenos, el último paso es integrar todo lo anterior en un sistema de realidad virtual: Oculus Rift DK2.

Para ello hubo que adaptar el software a la estructura empleada por Oculus, en la cual hay varias partes diferenciadas: Una primera parte en la que se inicializan las variables y se crea la look-up table, una segunda parte, la principal, llamada *update()*, que corresponde al núcleo del bucle principal y donde se encuentra la mayor parte del código (lectura de ficheros, procesamiento de imágenes…) y finalmente *draw()*, donde se realiza la representación fosfénica.

Además de adaptar el software a la estructura de Oculus, hubo que cambiar el sistema de referencia de la imagen a la hora de hacer la representación mediante fosfenos, ya que el sistema de referencia que emplea Oculus es distinto al que se empleó anteriormente, por lo la imagen se encontraba invertida utilizando el sistema de referencia anterior (figura 4.8)

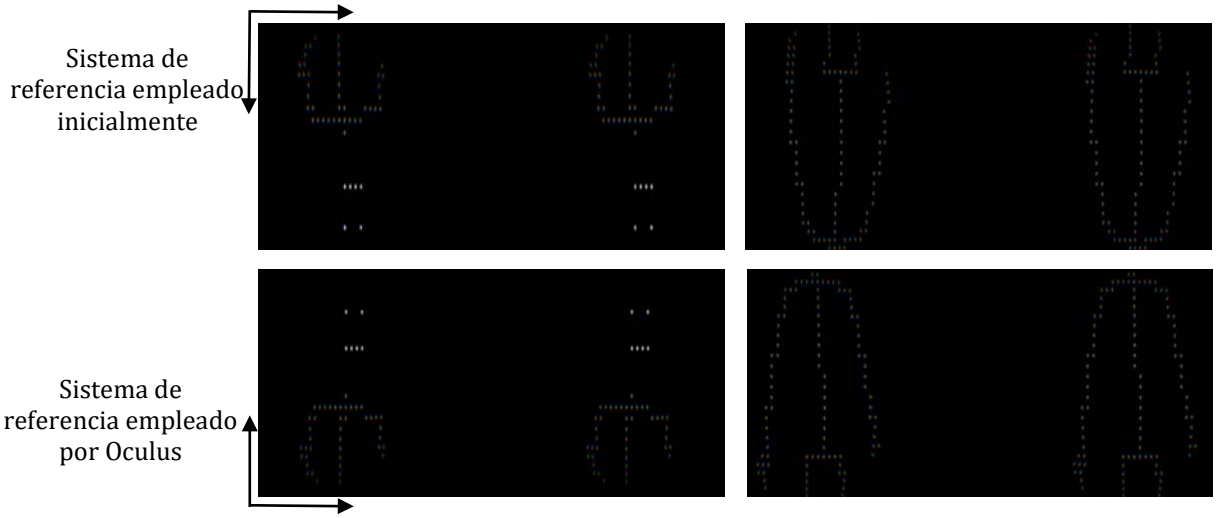


Figura 4.8. Ejemplos de representación en Oculus antes (arriba) y después (abajo) de cambiar el sistema de referencia de la imagen

Capítulo

5

EXPERIMENTACIÓN

A continuación se van a mostrar diferentes casos en cuanto a la representación tanto de la cara como del esqueleto, para distintas resoluciones del mapa de fosfenos con el fin de poder sacar conclusiones sobre qué tipo de representación es la más adecuada en cada caso, o si hay uno o varios tipos de representación que destaquen por encima del resto en cuanto a la información útil que proporcionan al usuario y la claridad de lo que este será capaz de percibir mediante la prótesis, ambos dos, aspectos básicos de la representación mediante fosfenos.

5.1 EXPERIMENTACIÓN CARA

En este apartado se van a mostrar diferentes ejemplos de los tres tipos de representación de la cara que se han propuesto anteriormente en este documento, en función de las tres resoluciones propuestas.

5.1.1 EXPERIMENTACIÓN CARA CON RESOLUCION BAJA

Considerando un mapa de fosfenos de baja resolución, en primer lugar se va a mostrar la representación en el que se muestran tanto ojos y boca, como la cara:

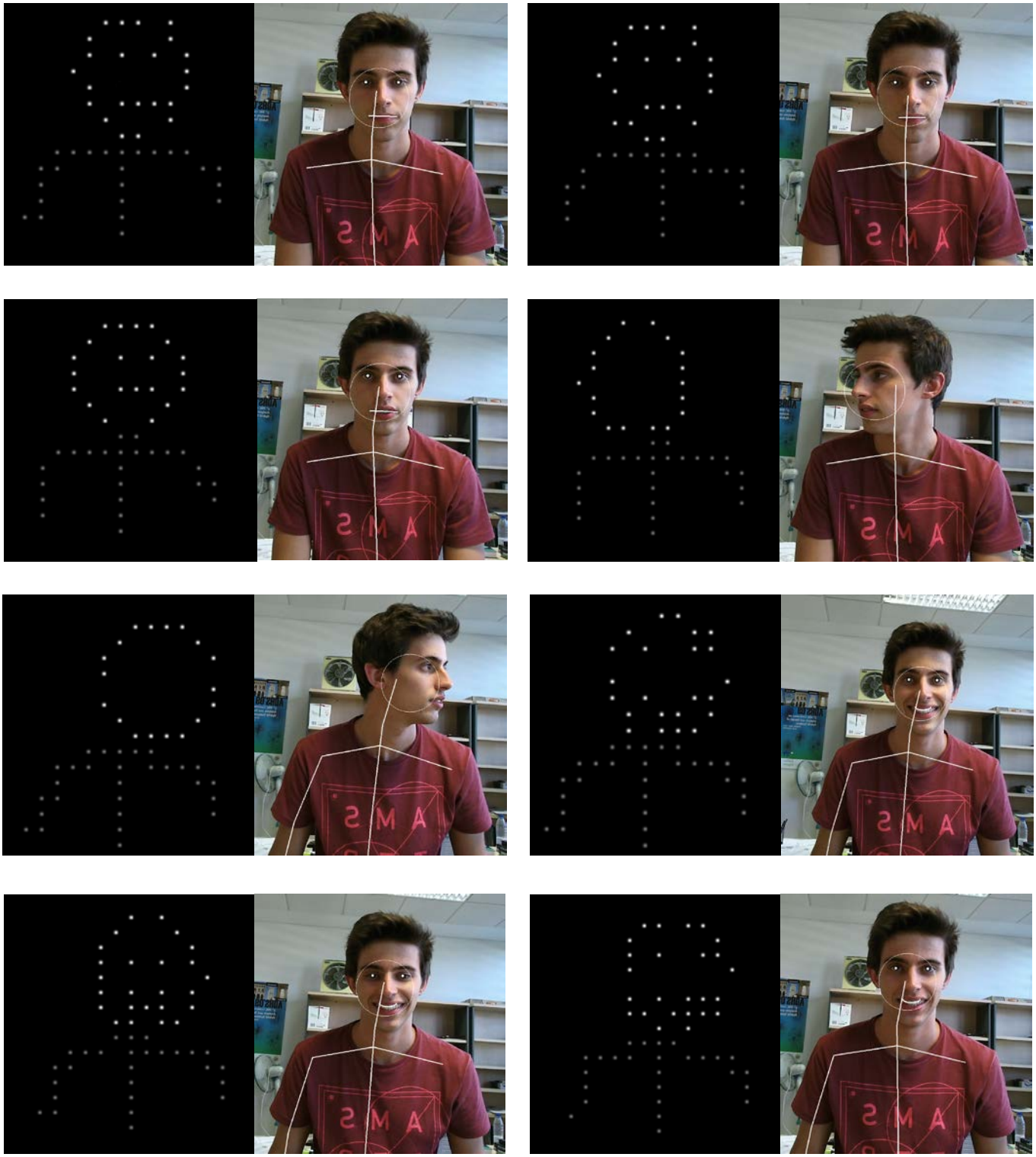


Figura 5.1. Ejemplos de representación de la cara tipo 1 con baja resolución

En estas imágenes de la figura 5.1 se aprecia como en varias ocasiones, en especial en el último caso, se confunde la boca con el contorno de la cara. Además de que ésta difiere bastante de ser una circunferencia en ciertos casos como en las dos primeras imágenes (primera fila)

Ahora se va a realizar la experimentación con el segundo tipo de representación, es decir, se mostrarán ojos, boca y cejas:

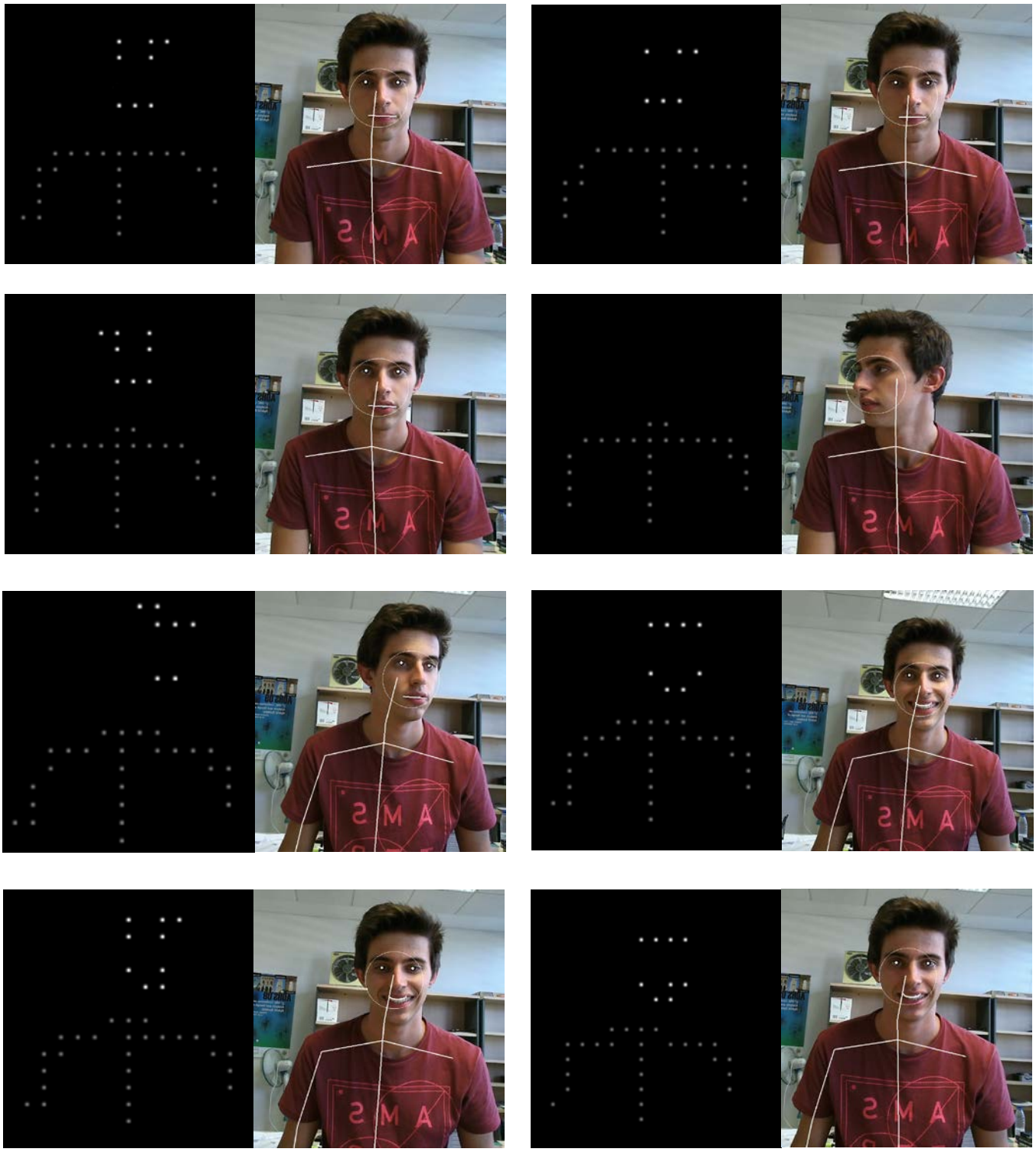


Figura 5.2. Ejemplos de representación de la cara tipo 2 con baja resolución

En las imágenes de la figura 5.2 se puede ver que hay muchas ocasiones en el que los ojos se confunden con las cejas.

Para acabar la experimentación con baja resolución, se van a mostrar las diferentes imágenes correspondientes al tercer tipo de representación, en la que solo se representan ojos y boca:

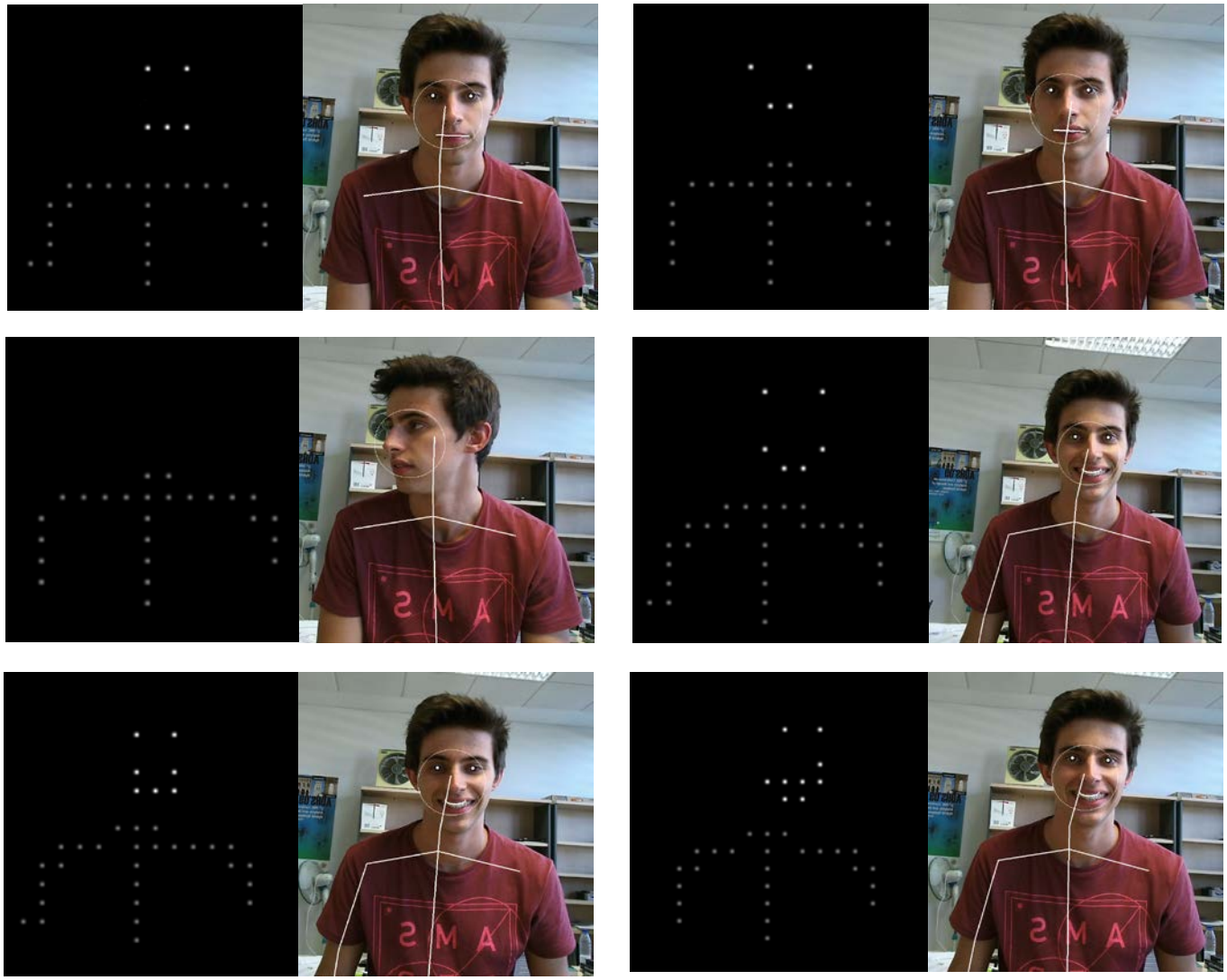


Figura 5.3. Ejemplos de representación de la cara tipo 3 con baja resolución

Como se puede ver, la última imagen de la figura 5.3 es la única en la que puede haber algún tipo de confusión en la representación, ya que resulta complicado interpretar la sonrisa de la persona detectada.

Para terminar este apartado, se podría concluir que para resoluciones bajas, la mejor forma de representar la cara es mostrando únicamente ojos y boca (tipo 3). Es la menos confusa de las tres, y hay que tener en cuenta que a bajas resoluciones la representación, en muchas ocasiones, carece de la claridad suficiente, por lo que es más sencillo cometer errores de interpretación

5.1.2 EXPERIMENTACIÓN CARA CON RESOLUCION MEDIA

En este apartado se va realizar experimentaciones similares, pero utilizando una resolución media del mapa de fosfenos:

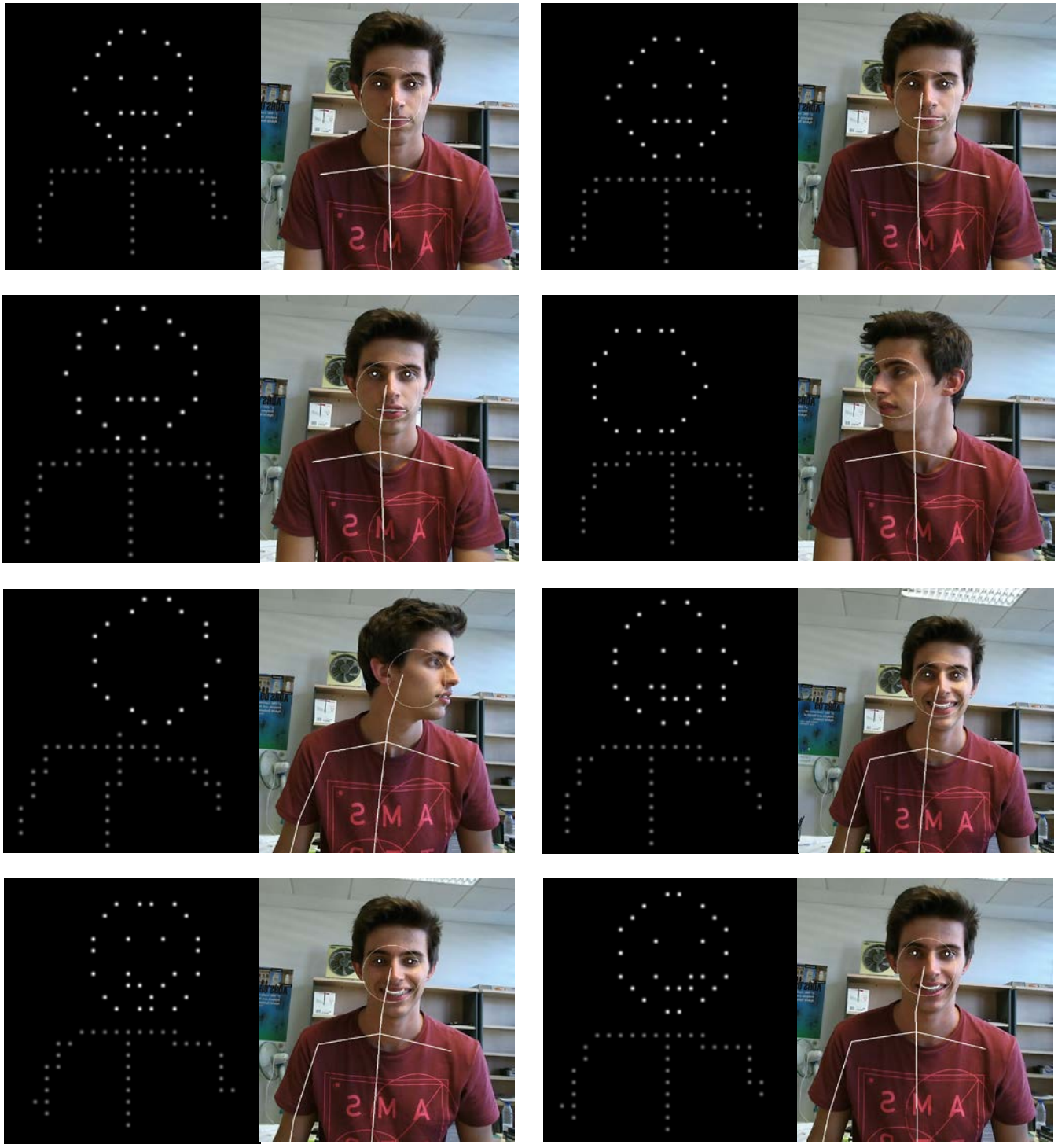


Figura 5.4. Ejemplos de representación de la cara tipo 1 con resolución media

En el primer ejemplo de la figura 5.4 no hay confusión entre boca y contorno de la cara, al contrario de lo que ocurría en el correspondiente a baja resolución (figura 5.1)

En el último ejemplo de la figura 5.4 ya no hay confusión entre la boca y el contorno de la cara, algo que si ocurría en el correspondiente ejemplo a baja resolución (figura 5.1).

Con este tipo de representación y con esta resolución, vemos que no hay ningún problema en cuanto a interpretación de todas las partes de la cara. Quizás en el penúltimo ejemplo, se podría llegar a confundir la boca con el contorno de la cara, pero no resulta imposible ni mucho menos interpretar la imagen correctamente.

Vista el primer tipo de representación, pasamos al segundo, en el que se mostrarán ojos, boca y cejas (figura 5.5):

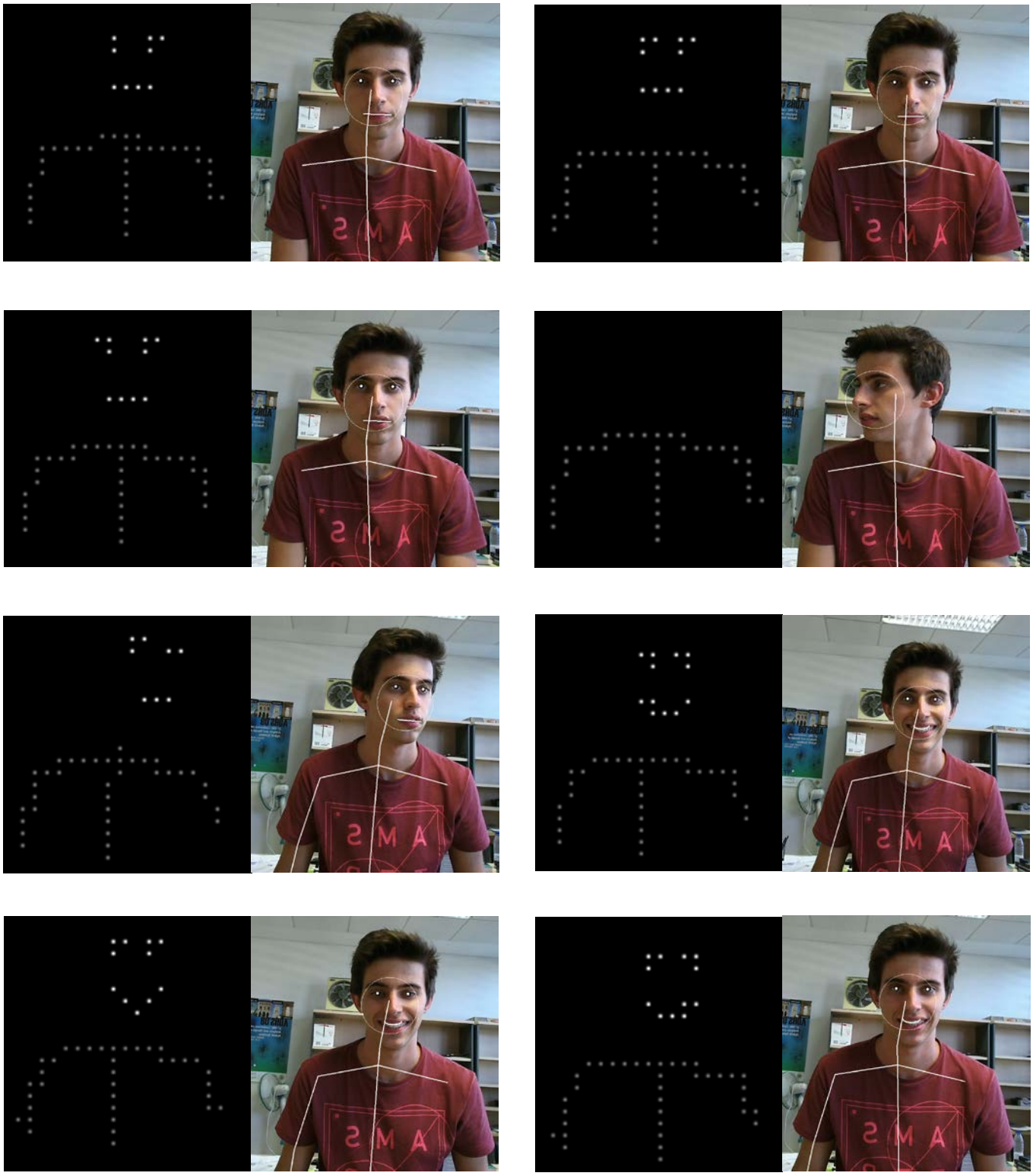


Figura 5.5. Ejemplos de representación de la cara tipo 2 con resolución media

En la quinta imagen de la figura 5.5 (tercera fila, primera columna) se ha reducido algo la confusión existente entre los ojos y las cejas que había en el correspondiente ejemplo a baja resolución (figura 5.2), aunque sigue habiéndola

En la sexta imagen de la figura 5.5 (tercera fila, segunda columna) ya no existe la confusión que había entre los ojos y las cejas en el correspondiente ejemplo a baja resolución (figura 5.2).

En el último ejemplo de la figura 5.5 (cuarta fila, segunda columna) ya no existe la confusión que había entre los ojos y las cejas en el correspondiente ejemplo a baja resolución (figura 5.2).

Se puede ver cómo, con el hecho de aumentar la resolución del mapa de fosfenos se ha conseguido solucionar la mayoría de los problemas de interpretación de la representación que había en el caso de baja resolución.

Para finalizar este apartado, se va a realizar la experimentación para el último de los tipos de representación propuestos (figura 5.6):

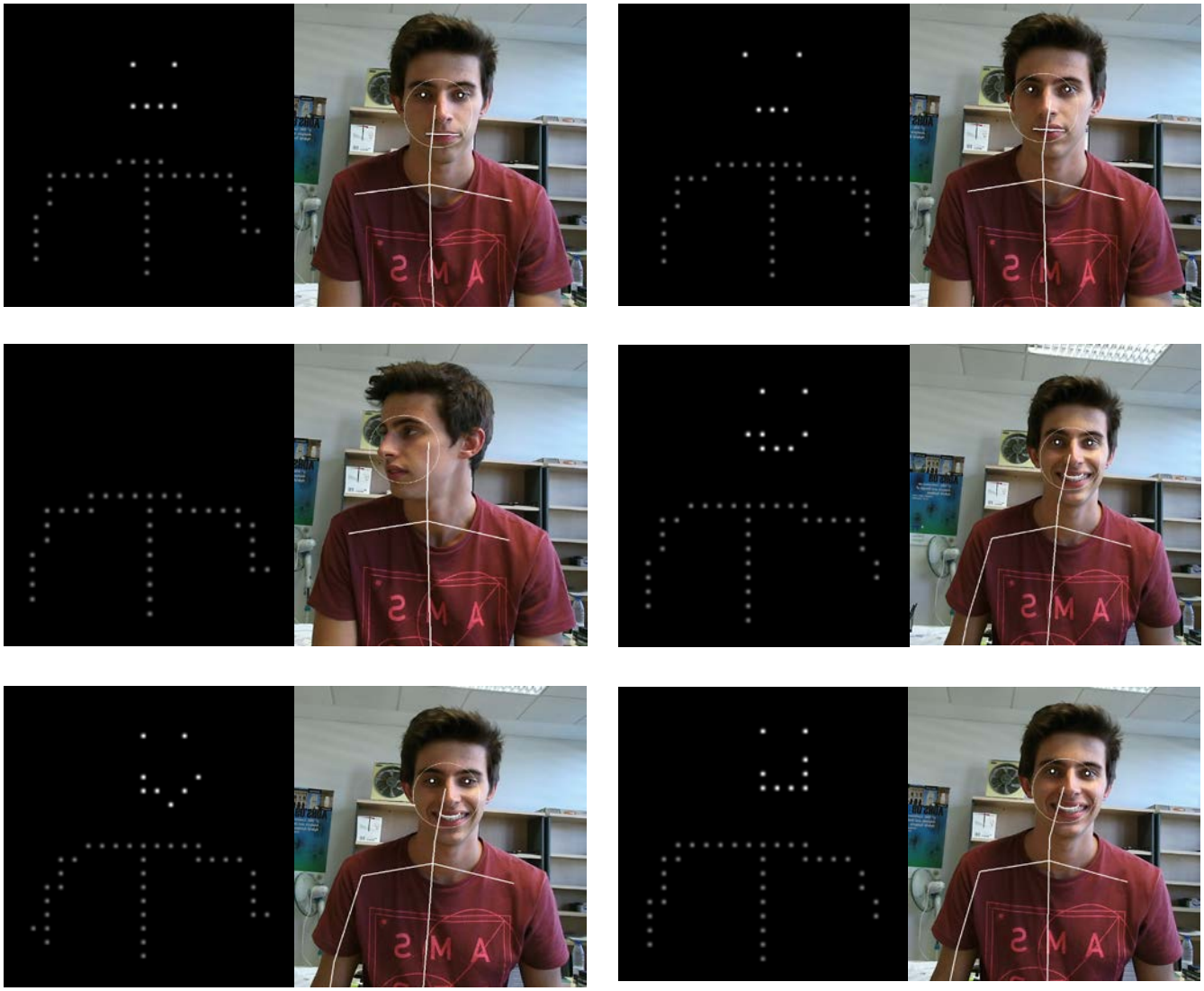


Figura 5.6. Ejemplos de representación de la cara tipo 3 con resolución media

Se ve como en la última imagen de la figura 5.6 se ha solucionado la confusión existente en el correspondiente ejemplo a baja resolución (figura 5.3), donde resultaba complicado interpretar la sonrisa de la persona detectada.

Con este tipo de representación y a esta resolución, se ve que es mucho más complicado encontrar problemas de interpretación.

Como conclusión, se puede decir que en el caso de utilizar la resolución media, se podría usar cualquiera de los tres tipos de representación. Ya que, pese a que tanto el primer y segundo tipo aportan algo más de información, hay algún caso en el que puede haber confusión (aunque a esta resolución son los menos), mientras que utilizando el tercer tipo, no habría confusión alguna, pero por el contrario se perdería algo de información.

5.1.3 EXPERIMENTACIÓN CARA CON RESOLUCION ALTA

Para finalizar la experimentación de la representación de la cara se van a mostrar diferentes ejemplos de los tres tipos de representación utilizando la máxima resolución propuesta del mapa de fosfenos. Se comenzará por la representación tipo 1:

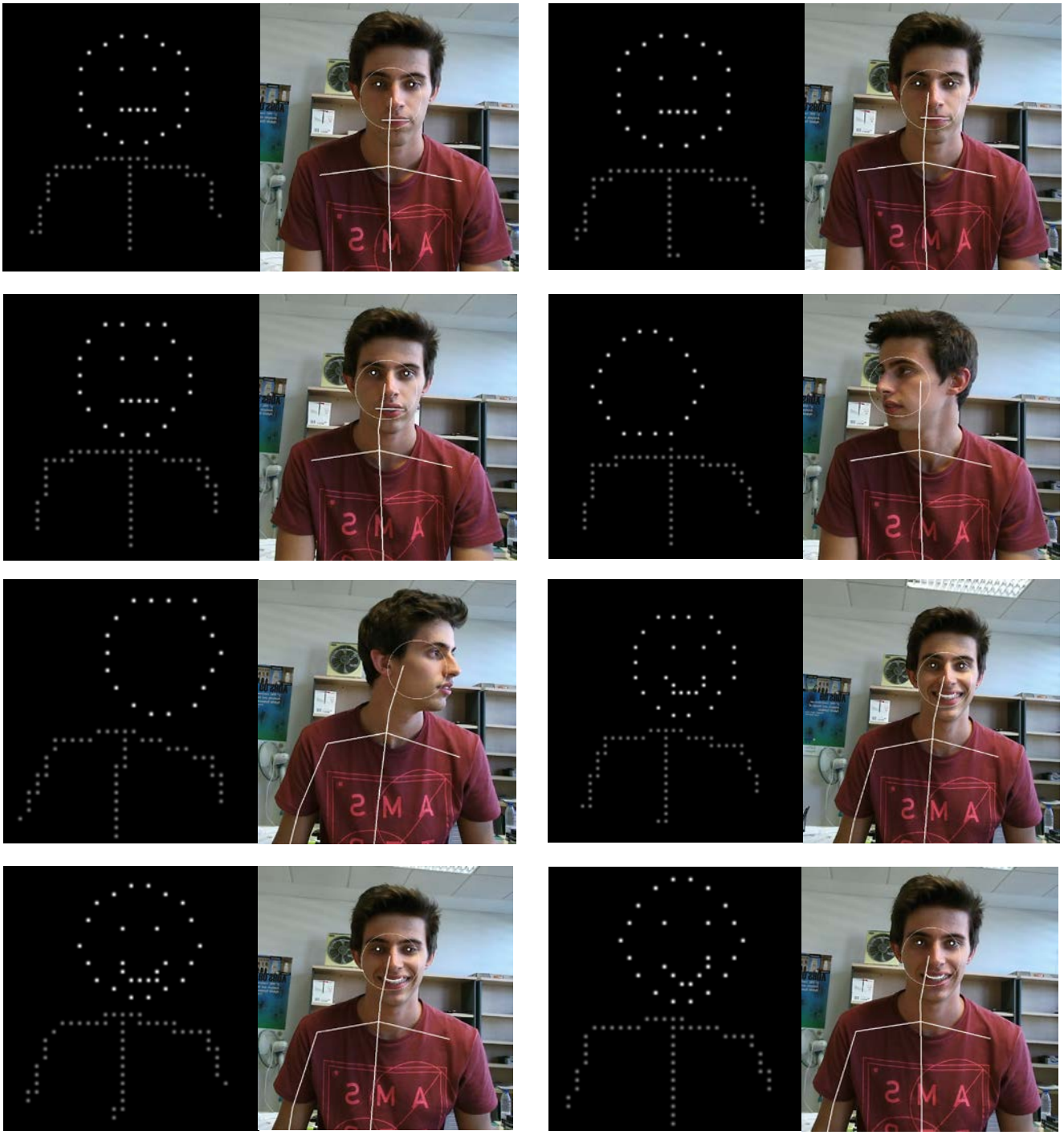


Figura 5.7. Ejemplos de representación de la cara tipo 1 con resolución alta

En la penúltima imagen (cuarta fila, primera columna) de la figura 5.7 se puede apreciar fácilmente la boca sonriendo, al contrario de lo que ocurría en la imagen correspondiente a resolución media (figura 5.4), donde la representación era algo confusa.

Utilizando resolución alta, obtenemos una representación en la que prácticamente no hay problemas de interpretación, ni confusión entre la boca y el contorno de la cara, como sí ocurría a resoluciones más bajas.

Ahora se va a realizar la misma experimentación, utilizando la representación tipo 2:

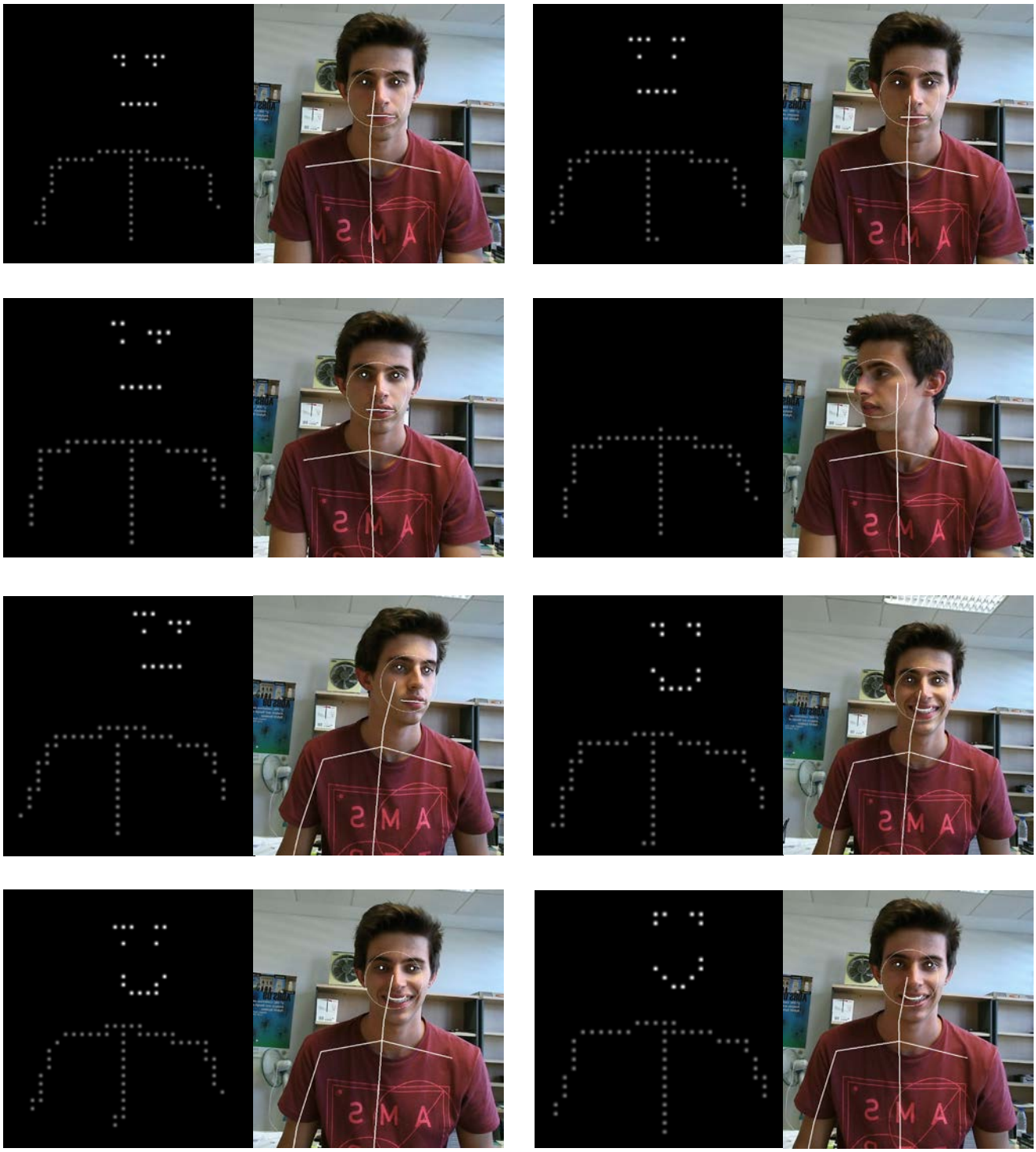


Figura 5.8. Ejemplos de representación de la cara tipo 2 con resolución alta

En la quinta imagen (tercera fila, primera columna) de la figura 5.8 se puede ver que ya no existen problemas de interpretación debidos a una representación confusa, como sí ocurría para la imagen correspondiente a resoluciones más bajas.

Como se puede observar en las diferentes imágenes expuestas anteriormente, a resolución alta, no hay prácticamente ningún problema de interpretación en la representación de tipo 2, por lo que es perfectamente viable su uso.

Por último se va a realizar la experimentación utilizando la representación tipo 3:

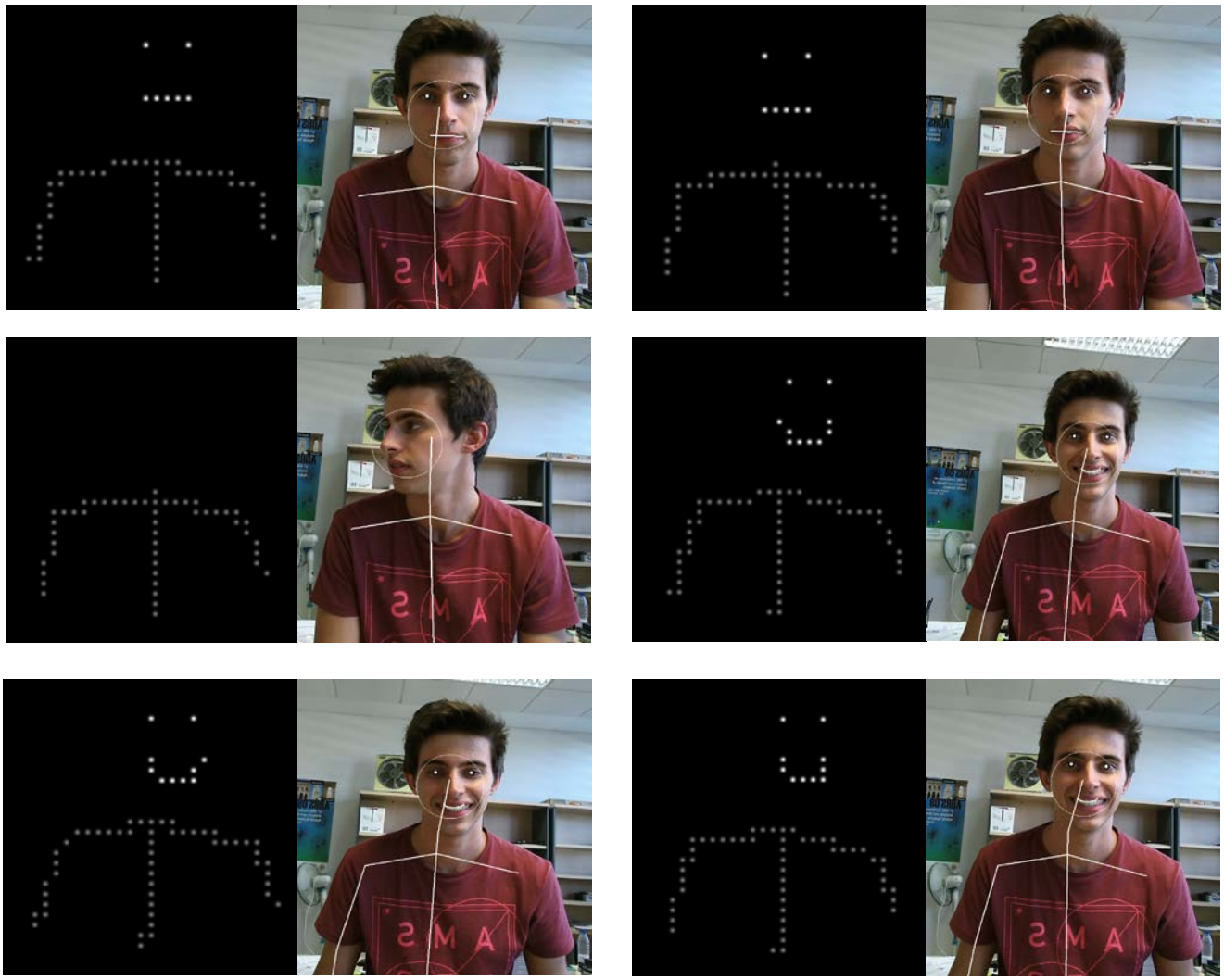


Figura 5.9. Ejemplos de representación de la cara tipo 3 con resolución alta

Como era de esperar, si para resolución media, no había habido problema alguno en cuanto a interpretaciones erróneas o confusiones utilizando este tipo de representación, aumentando la resolución tampoco se encuentran problemas en este aspecto.

Así pues, si utilizamos resoluciones altas, se podría utilizar cualquier tipo de representación, pero si hubiera que elegir una, sería el tipo 1, ya que mostrando el contorno de la cara además de los ojos y boca, se obtiene más información de la posición de la cara cuando la persona detectada no está mirando al usuario de la prótesis.

Finalmente, a modo de conclusión en cuanto a la representación de la cara, lo más apropiado sería utilizar la resolución media, atendiendo al compromiso entre claridad de la representación y dificultad técnica a la hora de aumentar la resolución de la prótesis. Teniendo en cuenta esto, si se desea no tener errores de interpretación en prácticamente ningún caso, aún a costa de perder algo de información que puede ser útil en algunas situaciones, la mejor opción sería usar la representación tipo 3. Si por el contrario, se quiere disponer de la máxima información posible aun pudiendo haber situaciones en las que esa información pueda ser confusa, lo más recomendable es utilizar la representación tipo 1. En cuanto a la de tipo 2, no es la más aconsejable ya que no aporta una información excesivamente útil y es más propensa a producir errores de interpretación.

5.2 EXPERIMENTACIÓN CUERPO

Ahora, se realizará un análisis similar en función de la resolución del mapa de fosfenos, pero para la representación del cuerpo. Por tanto, comenzamos con el de baja resolución (figuras 5.10 y 5.11):

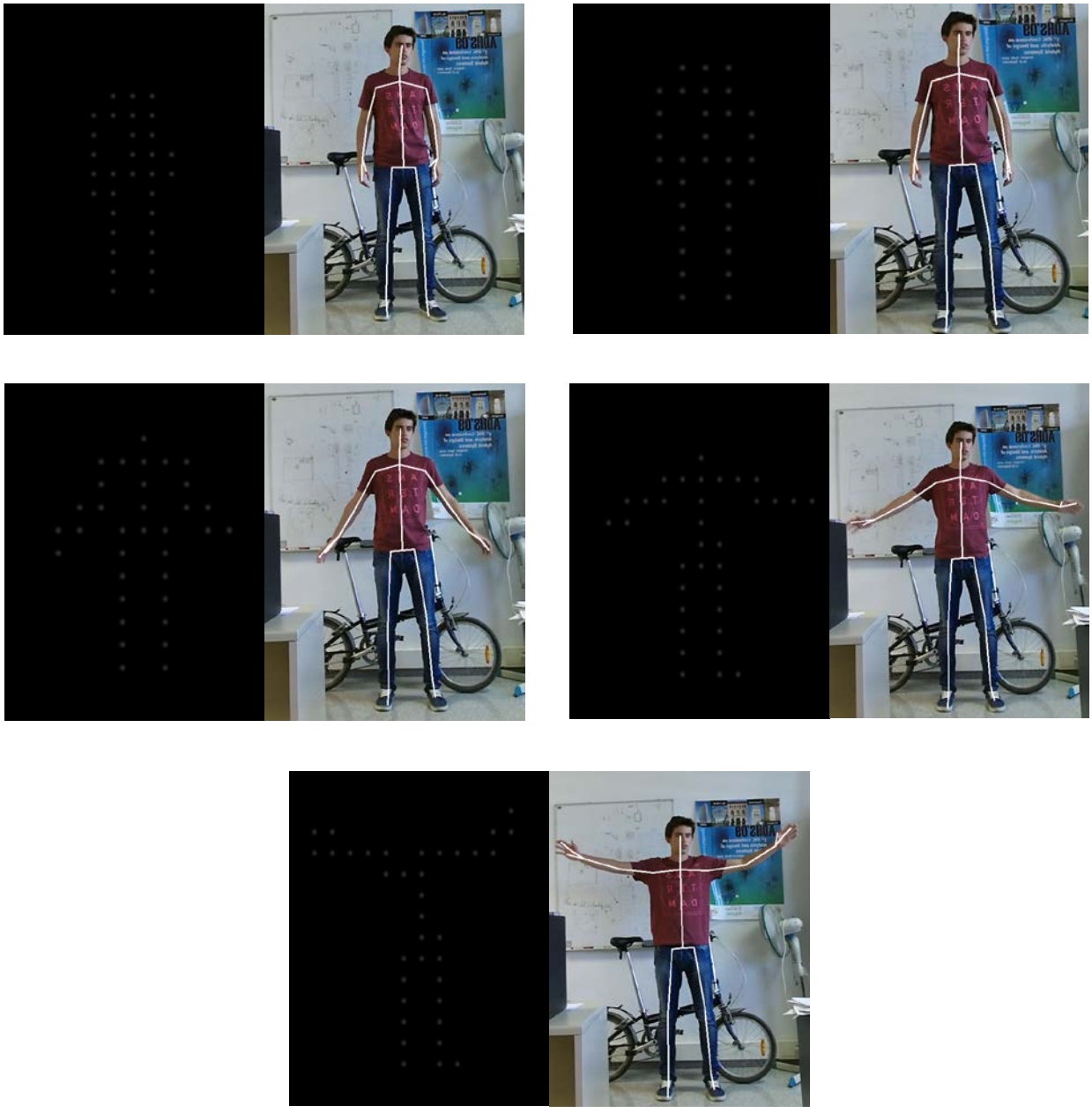


Figura 5.10. Ejemplos de representación del esqueleto con resolución baja

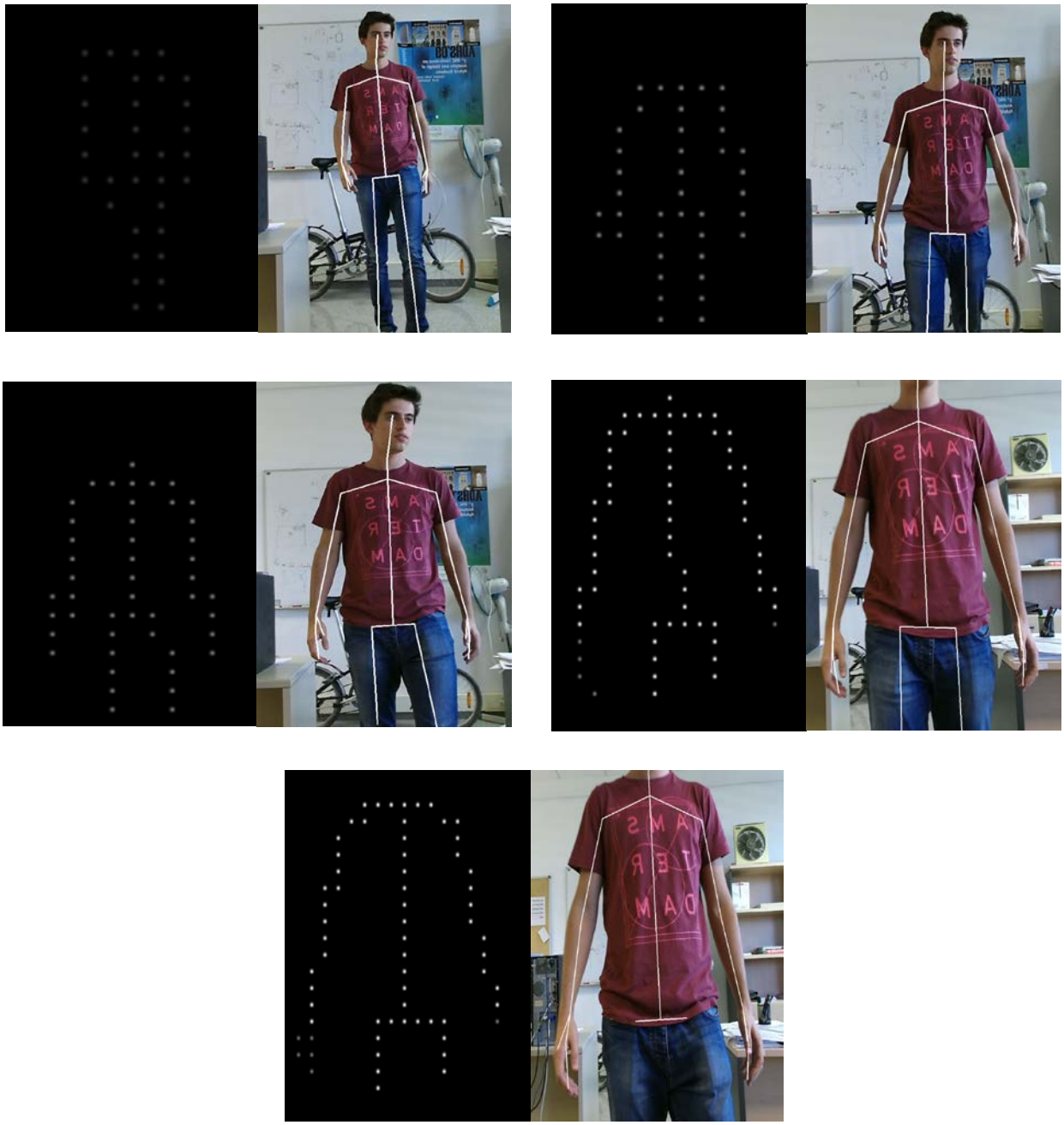


Figura 5.11. Ejemplos de representación del esqueleto con resolución baja

En el primer ejemplo de la figura 5.10, se aprecia como cuesta diferenciar los brazos del tronco en la representación fosfénica debido a la escasa resolución.

Sin embargo en la segunda imagen de la figura 5.10 se ve como ya se diferencia sin problema los brazos del tronco

Ahora se utilizará la resolución media (figuras 5.12 y 5.13):

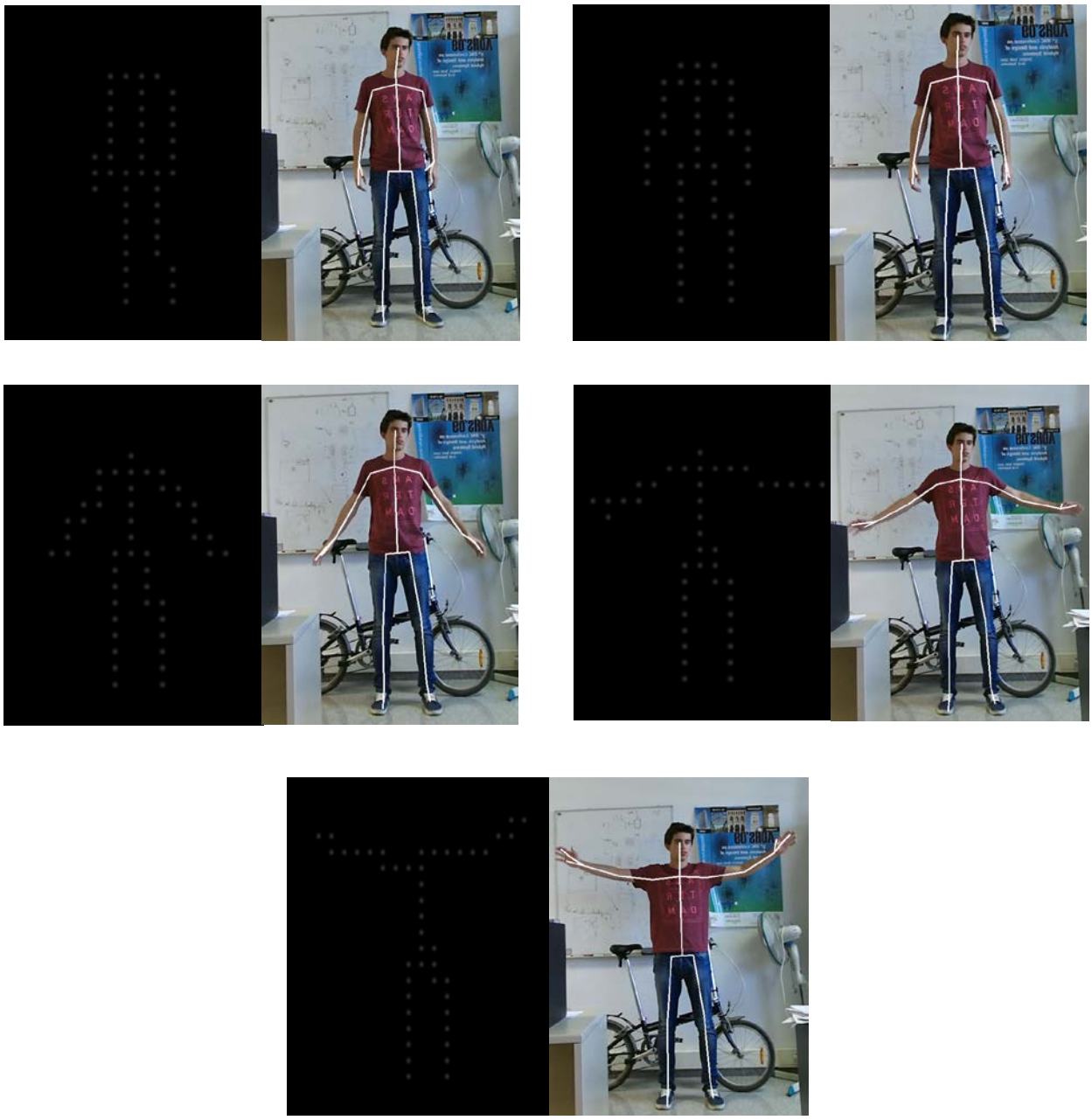


Figura 5.12. Ejemplos de representación del esqueleto con resolución media

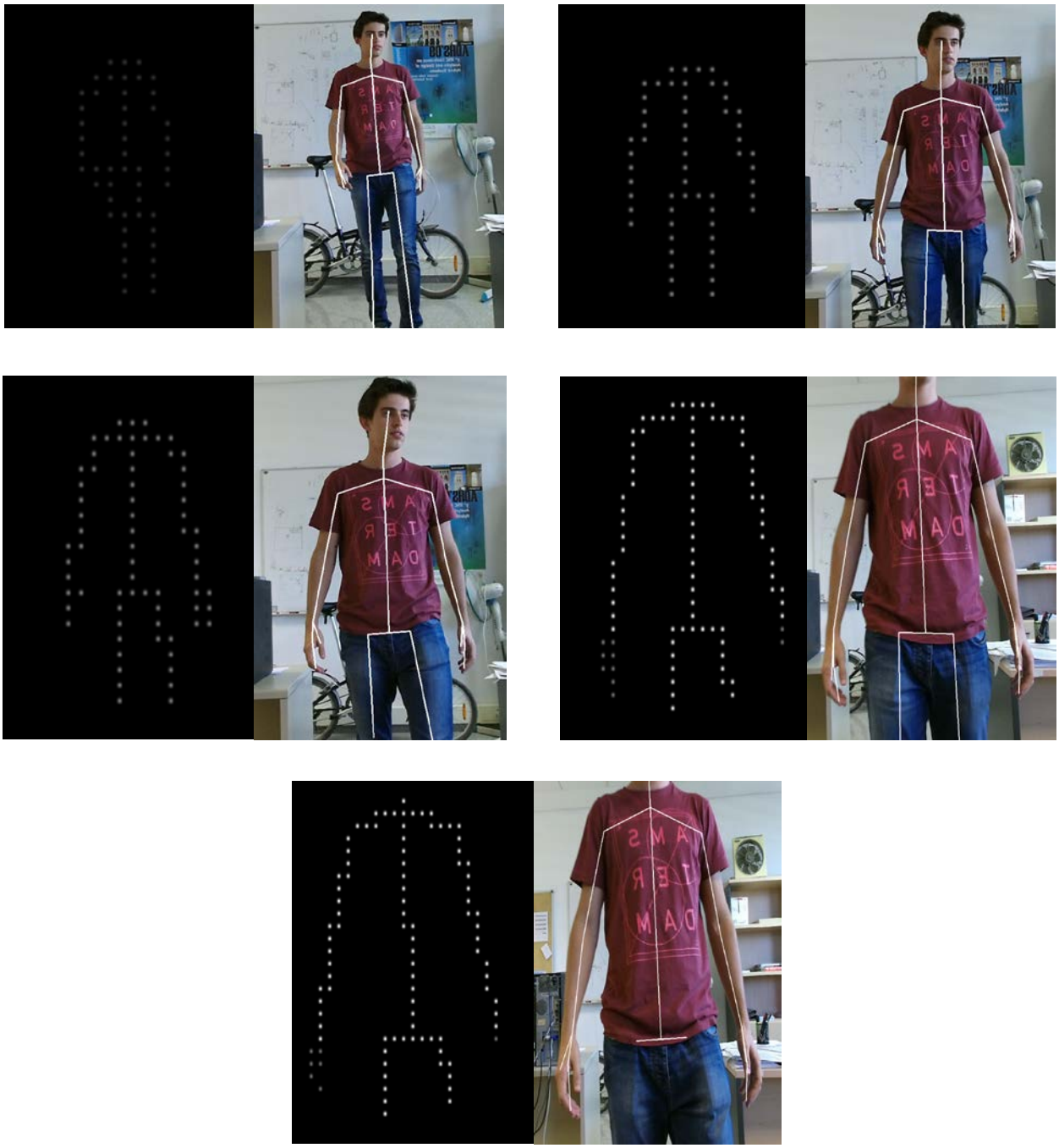


Figura 5.13. Ejemplos de representación del esqueleto con resolución media

En la primera imagen se aprecia, que al contrario de lo que ocurría con baja resolución, sí que se distinguen lo suficientemente bien los brazos del tronco, incluso cuando estos están muy pegados al resto del cuerpo.

Finalmente se utilizará la resolución alta, y se realizará la misma experimentación (figuras 5.14 y 5.15):

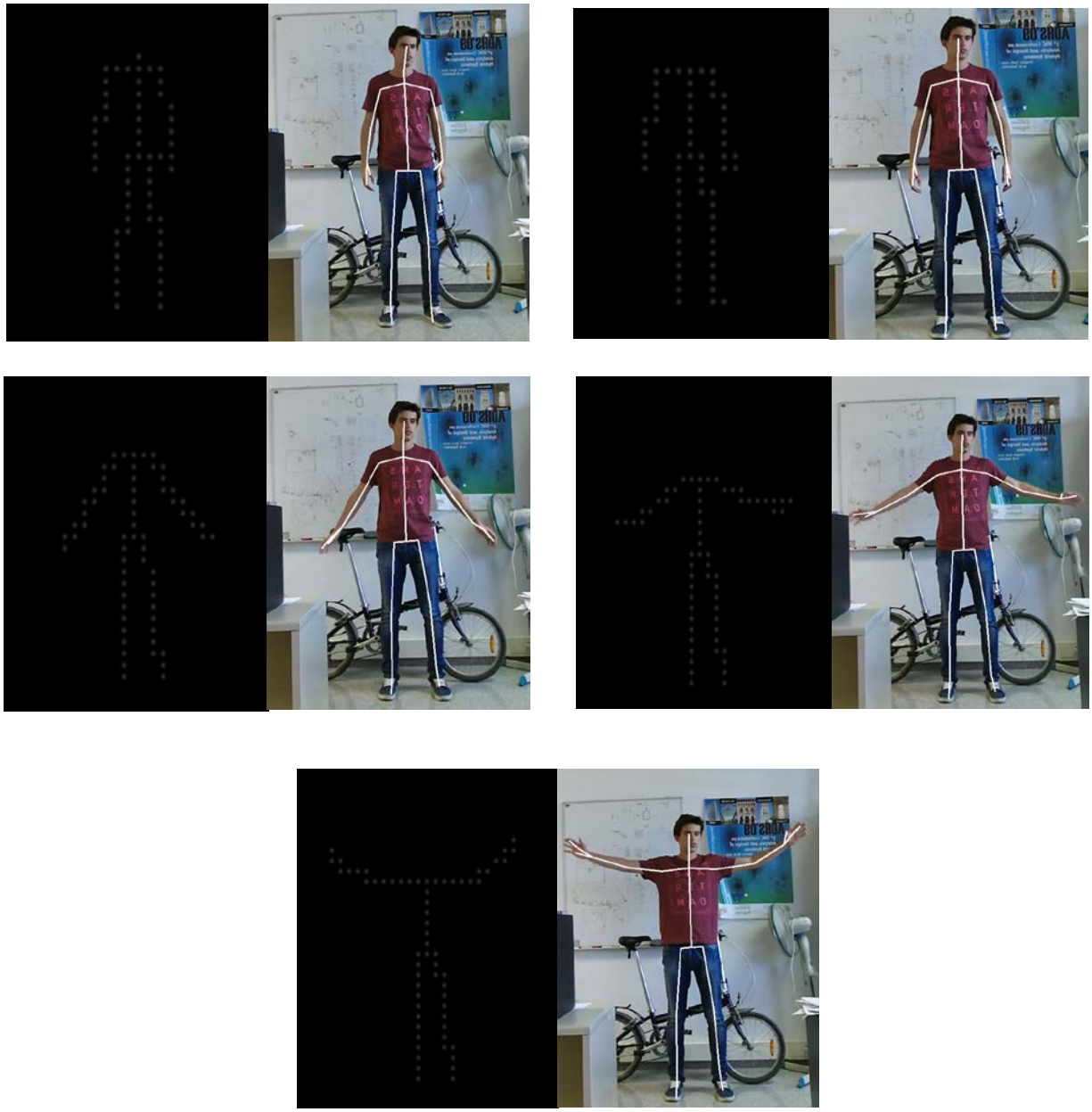


Figura 5.14. Ejemplos de representación del esqueleto con resolución alta

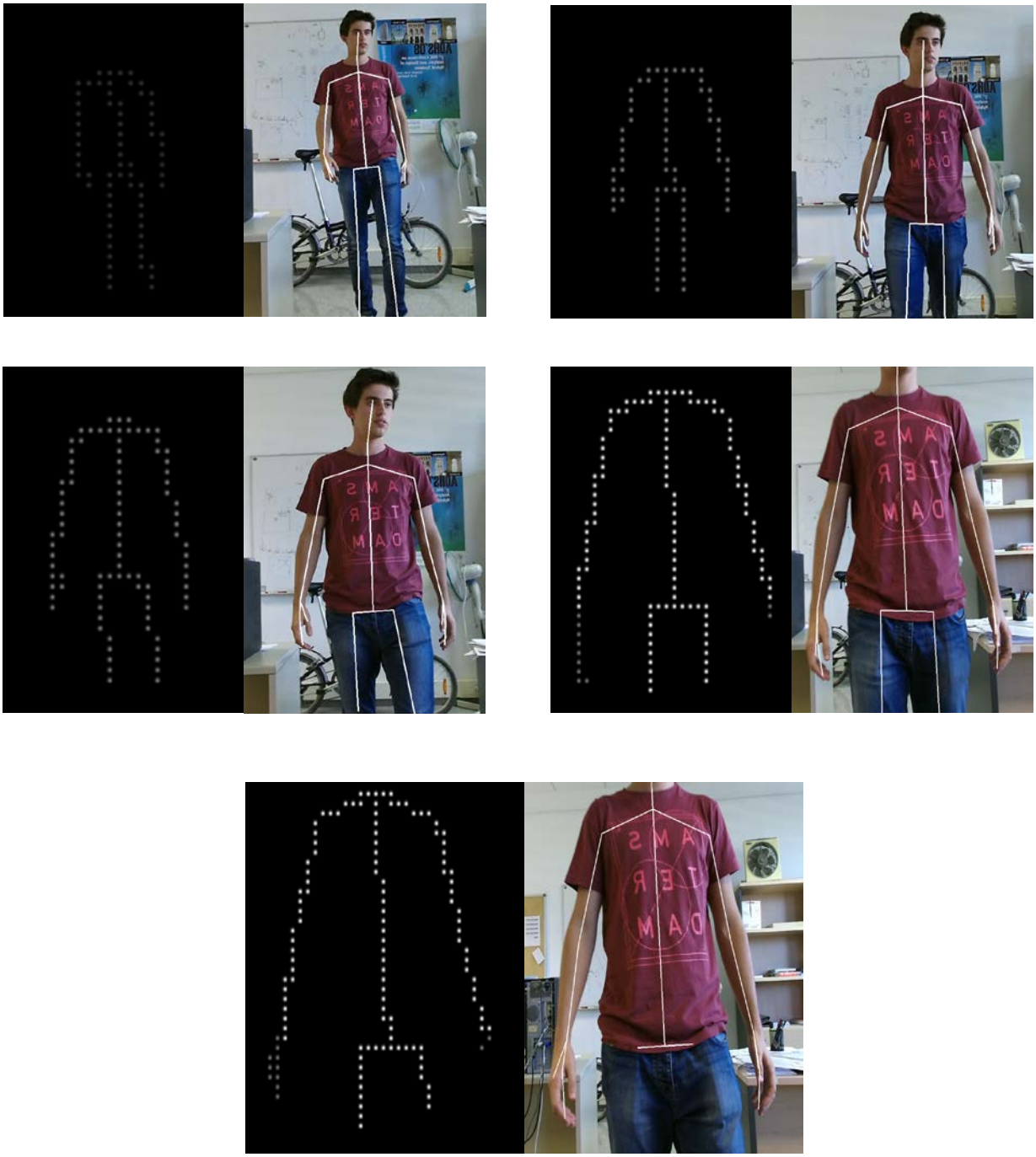


Figura 5.15. Ejemplos de representación del esqueleto con resolución alta

En lo que respecta al cuerpo, se puede decir que en general para cualquier resolución de las propuestas anteriormente, se puede apreciar suficientemente bien todos los aspectos importantes del esqueleto. Bien es cierto, que con la resolución baja, hay alguna ocasión en la que aparece cierta confusión entre algunas partes del cuerpo. Por lo tanto, se podría concluir que, atendiendo al compromiso existente entre la claridad en la representación y el coste técnico que supone aumentar el número de fosfenos de la prótesis, la resolución media es suficiente para representar el cuerpo.

Capítulo

6

CONCLUSIONES Y TRABAJO FUTURO

Para finalizar el trabajo vamos a exponer de forma resumida y ordenada los diferentes aspectos desarrollados a lo largo de la memoria. Al inicio del trabajo partimos de 2 dispositivos: Kinect v2 y Oculus Rift DK2. De este modo, y tras una familiarización con el estado del arte de las prótesis visuales, el primer paso ha sido la comprensión y manejo de las herramientas a utilizar. El siguiente paso ha sido el de realizar la implementación de aplicaciones para la detección y el reconocimiento de personas y su posterior representación icónica mediante fosfenos. Además, se han realizado experimentaciones para poder evaluar los diferentes tipos de representación, en función de los distintos mapas de fosfenos diseñados. El último paso ha sido la integración de todo lo anterior en un sistema de realidad virtual como es Oculus Rift DK2.

Como muchas veces ocurre en el desarrollo de proyectos relacionados con la investigación, el punto de finalización de un trabajo no está definido claramente. Es necesario fijar unos objetivos y dar el trabajo por terminado cuando éstos se cumplen, puesto que de lo contrario podría convertirse en un trabajo interminable. Sin embargo, es necesario realizar un ejercicio de exploración del trabajo futuro y considerar las posibles mejoras a aplicar. Concretamente, en este proyecto, se considera apropiado en primer lugar la realización de pruebas con usuarios, para poder evaluar de manera efectiva si la interpretación de la representación mediante fosfenos es adecuada, y en qué aspectos podría mejorarse dentro de las limitaciones existentes.

Pese a que ya se han incorporado algunas técnicas de reconocimiento facial, se cree apropiado también la utilización de más funcionalidades de reconocimiento de personas, ya que son técnicas en auge actualmente, y por tanto en constante evolución, por lo que puede ser de utilidad incorporarlas a las prótesis visuales.

Por último, en cuanto al sensor utilizado (Kinect v2), hay que ser consciente de las limitaciones que tiene, tales como que el rango de detección no es excesivamente amplio, o que en el momento que se mueve, comienzan a aparecer errores de detección. Bien es cierto, que algunos de estos problemas, podrían solucionarse utilizando algún software que sea capaz de lidiar con ellos, sin necesidad de

cambiar el sensor, aunque sería conveniente valorar la posibilidad de buscar sensores alternativos, para realizar la detección, sensores que, por ejemplo, sean menos voluminosos (otro de los hándicaps de Kinect), aunque esto obligaría a tener que desarrollar toda la parte de software que Microsoft nos pone a nuestra disposición al adquirir su producto.

Para terminar, solo queda añadir que se han cumplido los objetivos planteados al comienzo del proyecto. La sensación al finalizar este trabajo ha sido satisfactoria puesto que no se han encontrado barreras infranqueables, aunque el esfuerzo ha sido considerable. La expectativa ahora mismo, es que este trabajo sirva como inicio para posteriores investigaciones más profundas en nuevas técnicas de representación de imagen para prótesis visuales.

A

CALIBRACIÓN DE DISPOSITIVOS

Todo el proceso de calibración está recogido y ampliamente explicado en [BADIAS-2016]. En este anexo, solo se recogerán los aspectos más importantes del mismo.

El proceso de calibración nos permite obtener los parámetros intrínsecos que modelan las cámaras de los dispositivos y que son necesarios para relacionar, junto con los parámetros extrínsecos, los puntos de una escena 3D con sus puntos homónimos en la imagen.

Para calibrar las cámaras se supone un modelo pinhole donde la lente se ajusta al modelo proyectivo más unos coeficientes de distorsión radial. Un punto en el espacio se relaciona con un punto en la imagen no distorsionada con la matriz P:

$$u_{nd} = P(X, \theta) \qquad \text{(ecuación A.1)}$$

Donde u_{nd} representa las coordenadas de un punto en la imagen no distorsionada, X son las coordenadas de un punto en el espacio y _ representa los parámetros de la cámara ($\theta_{intr,}\theta_{extr}$). Los parámetros intrínsecos están asociados a la óptica de la cámara y a la geometría del sensor, mientras que los extrínsecos fijan la posición y orientación de la cámara en el espacio. La ecuación extendida puede verse en la ecuación A.2.

$$\begin{bmatrix} u \\ v \\ s \end{bmatrix} = \lambda \begin{bmatrix} \frac{f}{d_x} & 0 & c_x \\ 0 & \frac{f}{d_y} & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad \text{(ecuación A.2)}$$

$$u_{nd} = \begin{bmatrix} \frac{u}{s} \\ \\ \frac{v}{s} \end{bmatrix} \qquad \text{(ecuación A.3)}$$

Donde λ es el factor de escala y la matriz P [3x4] se expresa como P = K [R|t].
K es la matriz intrínseca y [R|t] representa la matriz extrínseca. La matriz extrínseca de la cámara transforma los puntos tridimensionales del espacio en puntos tridimensionales en la referencia de la cámara. Explica la rotación (R) y traslación (t) para cada punto desde sus coordenadas originales hasta las coordenadas de la cámara. La matriz intrínseca K contiene información sobre el centro de proyección (c_x, c_y), el tamaño de los pixeles (d_x, d_y) y la distancia focal (f), que se utiliza para mapear los puntos 3D en la referencia de la cámara sobre las coordenadas pixélicas. El conjunto de matrices hace posible la transformación de puntos 3D a píxeles de la imagen.

La corrección que se aplica para tener en cuenta la distorsión (radial y tangencial) se modela con la ecuación A.4.

$$u_d = (1 + k_1 r^2 + k_2 r^4 + k_5 r^6) u_{nd} + du \qquad \text{(ecuación A.4)}$$

Siendo $r^2 = u_{nd1}^2 + u_{nd2}^2$ y du la componente tangencial de la distorsión

$$du = \begin{bmatrix} 2k_3 u_{nd1} u_{nd2} + k_4 (r^2 + 2u_{nd1}^2) \\ k_3 (r^2 + 2u_{nd2}^2) + 2k_4 u_{nd1} u_{nd2} \end{bmatrix} \qquad \text{(ecuación A.5)}$$

El proceso de calibración consiste en la minimización del error de reproyección de cada punto *j* observado en la imagen *i* para estimar la posición real de los n puntos, y los parámetros extrínsecos e intrínsecos de las m imágenes. Los parámetros intrínsecos se consideran fijos para todas las imágenes, puesto que la cámara empleada será la misma.

Se ha empleado la toolbox de calibración para Matlab de Jean-Yves Bouguet [BOUGUET-2004] tomando imágenes de un patrón plano de tablero de ajedrez. Marcando las esquinas de cada cuadro del patrón en cada imagen y conociendo el tamaño real de los cuadros, el código implementado por Bouguet permite obtener una estimación de los parámetros de la cámara. A continuación se adjuntan los parámetros intrínsecos estimados de cada cámara.

A.1 CÁMARA IR DE KINECT

Según uno de los empleados de Microsoft [SIRIGNANO-2015] cada sensor Kinect es calibrado antes de salir al mercado y el propio dispositivo almacena sus parámetros intrínsecos. Mediante el SDK somos capaces de acceder a dicha información. Los parámetros se recogen en la tabla A.1.

Parámetro	Valor
Distancia focal en X	365.353 píxeles
Distancia focal en Y	365.353 píxeles
Centro de proyección en X	261.198 píxeles
Centro de proyección en Y	206.434 píxeles
Distorsión radial 2°orden	0.084298
Distorsión radial 4° orden	-0.268191
Distorsión radial 6° orden	0.101543

Tabla A.1. Parámetros intrínsecos según Microsoft para la cámara IR de Kinect

Para nuestro proceso de calibración se han tomado 18 imágenes (512 x 424 píxeles) del patrón en distintas posiciones (figura A.1). Ha sido necesario invertir las imágenes con respecto a su eje vertical, puesto que por defecto Kinect muestra las imágenes en modo espejo (está pensada para el uso doméstico de modo que el usuario se vea como en un espejo). El resultado de la calibración se recoge en la tabla A.2.

Las diferencias entre los parámetros obtenidos con la calibración de Bouguet y la de Microsoft son relativamente pequeñas, además de que en algunos casos los errores quedan dentro del intervalo de incertidumbre (fijado como 3 veces la desviación estándar del conjunto de parámetros intrínsecos estimados a priori, leer el proceso de calibración de Bouguet para profundizar). Microsoft no tiene en cuenta las desviaciones tangenciales, mientras que Bouguet sí lo hace, lo que puede ser motivo de esas diferencias entre los valores. Aun así, los procesos de calibración son estimaciones, por lo que siempre se cometerán errores puesto que no conocemos la realidad. Además, se trata de encontrar unos parámetros para modelar las cámaras y sus posiciones, pero en ocasiones diferentes soluciones pueden resolver el sistema del mismo modo.

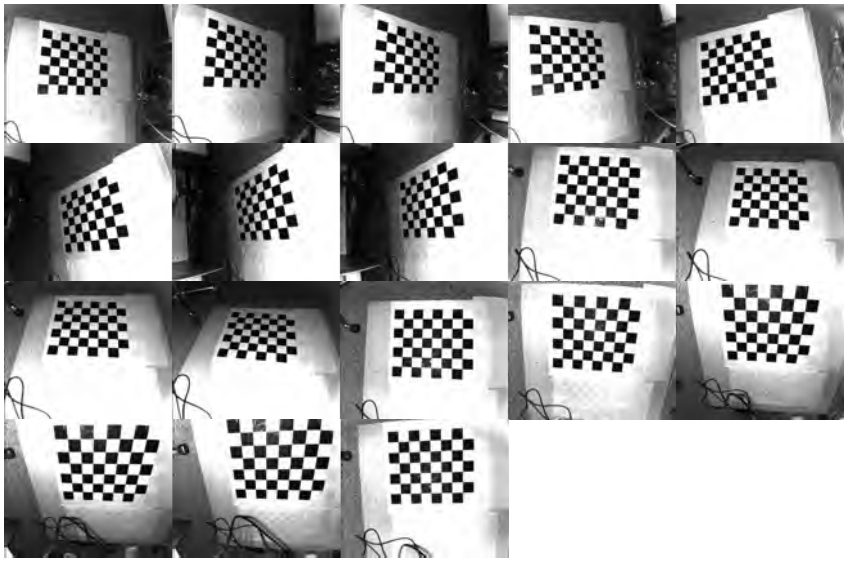


Figura A.1. Imágenes tomadas con la cámara de infrarrojos de Kinect para el proceso de calibración

Parámetro	Valor (± incertidumbre)	Desviación (val. absoluto)	Error relativo
Dist. focal en X	365.487 ±1.582 pix.	0.133	0.0364
Dist. focal en Y	364.071 ±1.597 pix.	1.282	0.3509
Centro imagen X	251.023 ±1.637 pix.	10.175	3.8955
Centro imagen Y	207.602 ±1.664 pix.	1.168	0.5658
D. radial 2° ord. K_1	0.07707 ±0.00780	0.00722	8.5649
D. radial 4° ord. K_2	-0.20023 ±0.01337	0.06796	25.3401
D. radial 6° ord. K_5	0.00000 ±0.00000	0.10154	100
D. tangencial K_3	-0.00086 ±0.00164	N.d.	N.d.
D. tangencial K_4	0.00109 ±0.00133	N.d.	N.d.
Error pixel X	0.345 píxeles	N.d.	N.d.
Error pixel Y	0.336 píxeles	N.d.	N.d.

Tabla A.2. Parámetros intrínsecos según la calibración llevada a cabo por la cámara IR de Kinect, desviación con respecto a la calibración de Microsoft y error relativo con respecto a los valores de Microsoft

A.2 CÁMARA RGB DE KINECT

Para la cámara de color el SDK de Kinect no da información de sus parámetros intrínsecos. Realmente se podría haber obviado la calibración de esta cámara, puesto que aunque el SDK no proporcione los datos de calibración, sí permite transformar los puntos desde la imagen de color a la de profundidad, y a la inversa. Aun así, se ha calibrado esta cámara por si fuera necesario su uso en un futuro. Se han empleado 20 imágenes (1920 x 1080 píxeles) del patrón (figura A.2) y el resultado se recoge en la tabla A.3.

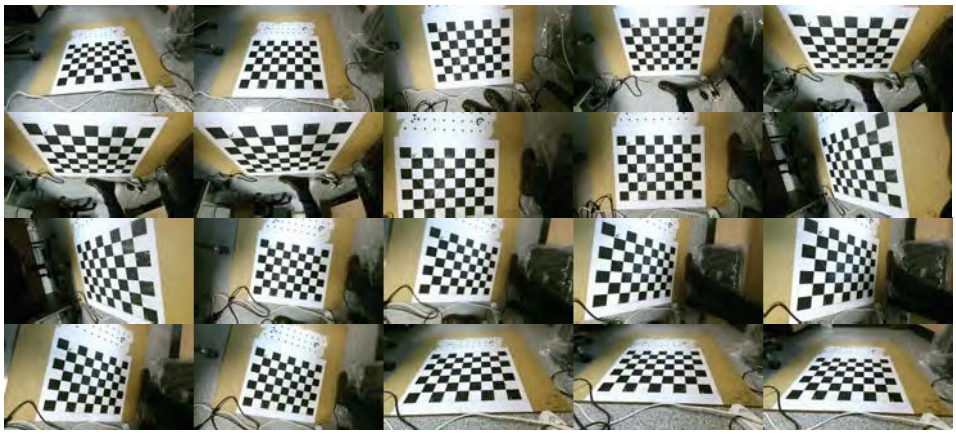


Figura A.2. Imágenes tomadas con la cámara RGB de Kinect para el proceso de calibración

Parámetro	Valor (± incertidumbre)
Distancia focal en X	1054.847 ± 1.795 pix.
Distancia focal en Y	1051.543 ± 1.849 pix.
Centro imagen X	986.119 ± 3.712 pix.
Centro imagen Y	541.931 ± 2.757 pix.
Distorsión radial 2° orden K_1	0.03365 ± 0.00457
Distorsión radial 4° orden K_2	-0.04255 ± 0.00554
Distorsión radial 6° orden K_5	0.00000 ± 0.00000
Distorsión tangencial K_3	-0.00073 ± 0.00087
Distorsión tangencial K_4	-0.00188 ± 0.00126
Error pixel X	0.857 píxeles
Error pixel Y	0.805 píxeles

Tabla A.3. Parámetros intrínsecos según la calibración llevada a cabo para la cámara RGB de Kinect

A.3 TRANSFORMACIÓN KINECT-OCULUS

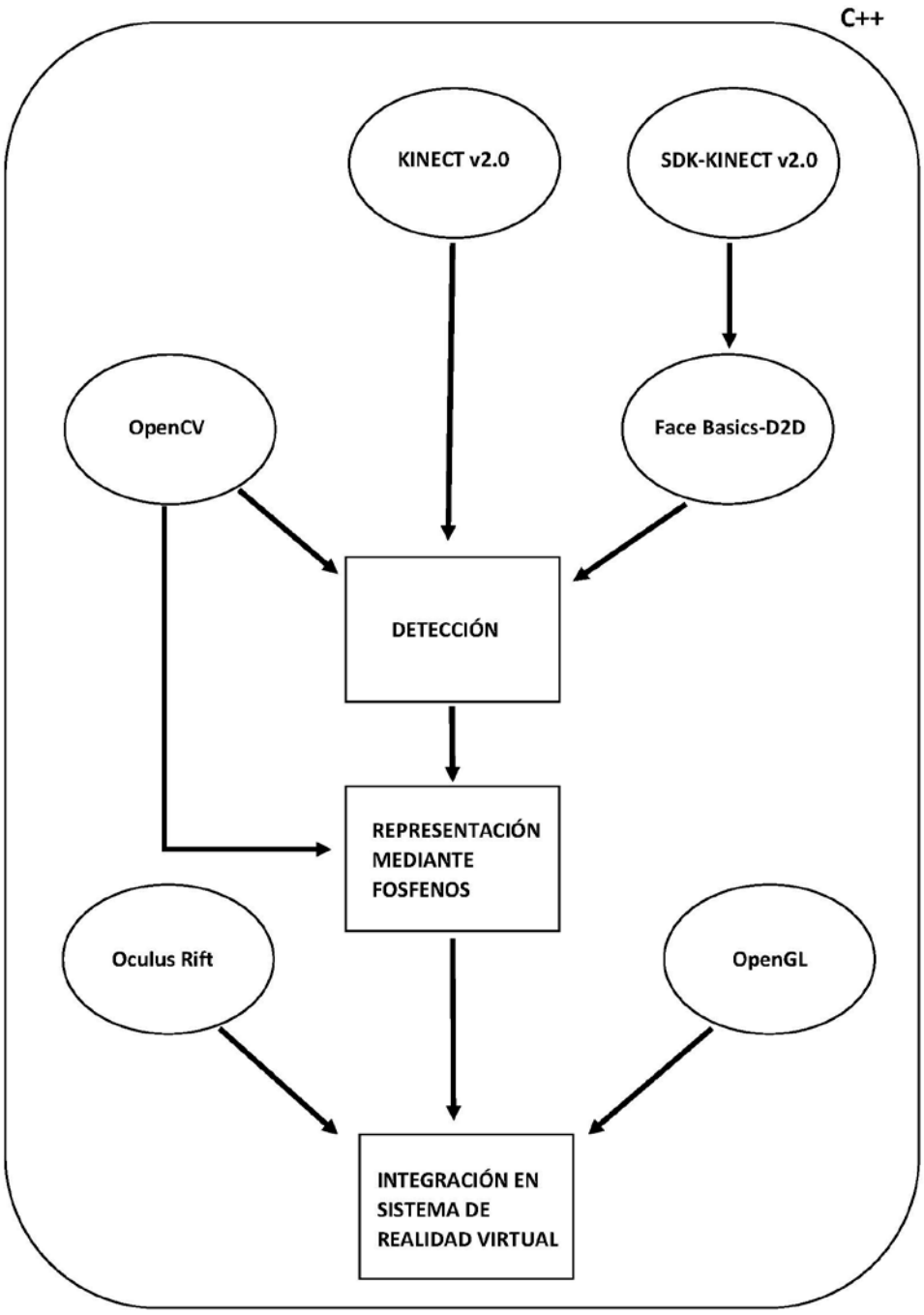
En cuanto al proceso de cálculo de la transformación Kinect-Oculus que existe en el montaje del simulador descrito en el capítulo 2, está ampliamente explicado en [BADIAS-2016], por lo que únicamente se va a mostrar el resultado final al que se ha llegado en dicho documento:

Kinect respecto Oculus (${}^{K}T_{O}$)	Oculus respecto Kinect (${}^{O}T_{K}$)
0.989 0.014 0.148 55.395 -0.008 0.999 -0.045 -97.135 -0.148 0.044 0.988 26.486 0 0 0 1	0.989 -0.008 -0.148 -51.599 0.014 0.999 0.044 95.087 0.148 -0.045 0.988 -38.746 0 0 0 1
Vector de rotación XYZ	Vector de rotación XYZ
2.562 8.502 -0.627 (grados)	-2.562 -8.502 0.627 (grados)
Traslación XYZ	Traslación XYZ
55.395 -97.135 26.486 (mm)	-51.599 95.087 -38.746 (mm)

Tabla A.4. Transformaciones entre las gafas de Oculus y la cámara de infrarrojos de Kinect tras la calibración.

B

ESQUEMA GENERAL DEL SOFTWARE



Como se puede ver en este esquema, partiendo del sistema de detección Kinect 2.0, del programa Face Basics-D2D perteneciente a la SDK de Kinect, y de algunas modificaciones del mismo, se realiza la detección.
Una vez realizada, mediante el uso de OpenCV y C++ (lenguaje en el que se trabaja en todo momento), utilizando el procesamiento adecuado, se obtiene la representación mediante fosfenos.
Finalmente, para poder integrarlo todo en un sistema de realidad virtual se necesita del sistema en si (Oculus Rift) y de las librerías OpenGL. Una vez adaptado el código al sistema de realidad virtual, obtenemos el simulador de prótesis visual.
En cuanto al software implementado, consta de tres programas principales y todos ellos son ficheros con extensión .sln:

- Face Basics-D2D modificado (*D:\Manuel\FaceBasics-D2D*), que se utiliza para realizar la detección de personas (cara y esqueleto). La detección se guarda en ficheros .raw para cada frame.
- Programa principal (*D:\Manuel\FaceBasics-D2D\ oculusSDKTestsEjemplos*): En este fichero se ha implementado en primer lugar la lectura de los ficheros que se han guardado en el programa anterior. También aquí es donde se ha implementado todo el procesamiento de los datos obtenidos, tanto para realizar la representación mediante líneas superpuestas a la imagen original que se utilizan como testeo de la detección, como para la realización de la representación mediante fosfenos.
- Integración en sistema de realidad virtual (*D:\isolateProjects\ oculusKinectInterface\source\oculusFaceRepresentation*): El último de los ficheros modificados. En él se encuentra la adaptación del software del programa principal para poder mostrar la representación mediante fosfenos en el sistema Oculus. Para ello, además de conocer la estructura software empleada por Oculus, hubo que cambiar el sistema de referencia a la hora de realizar la representación, ya que el utilizado en el programa principal, no coincidía con el del sistema de realidad virtual.

BIBLIOGRAFÍA

[AG-2015] Retinal Implant AG. Alpha ims features. http://www.retina-implant.de/en/patients/technology/default.aspx.

[ALADREN-2016] A Aladren, Gonzalo Lopez-Nicolas, Luis Puig, and Josechu J Guerrero. Navigation assistance for the visually impaired using rgb-d sensor with range expansion. *IEEE Systems Journal*, 10(3), 2016, pp: 922-932.

[ASUS] Asus. Dispositivo asus xtion pro. https://www.asus.com/es/3D-Sensor/Xtion PRO/.

[BADIAS-2016], Alberto Badias Herbera, Simulación de prótesis visual con sensor RGB-D, *Proyecto fin de Master en Ingeniería Biomédica, EINA, Universidad de Zaragoza*, 2016.

[BARNES-2013] Nick Barnes. An overview of vision processing in implantable prosthetic vision. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 1532–1535. IEEE, 2013.

[BOUGUET-2004] Jean-Yves Bouguet. Camera calibration toolbox for matlab. 2004

[BRINDLEY-1968] Giles S Brindley and WS Lewin. The sensations produced by electrical stimulation of the visual cortex. *The Journal of physiology*, 196(2):479-493, 1968.

[CASTANEDA-2011] Victor Castaneda and Nassir Navab. Time-of-flight and Kinect imaging, 2011.

[DAGNELIE-2011] Gislin Dagnelie. Visual prosthetics: physiology, bioengineering, rehabilitation. *Springer Science & Business Media*, 2011.

[DENNIS-2012] Wen Lik Dennis Lui, Damien Browne, Lindsay Kleeman, Tom Drummond, and Wai Ho Li. Transformative reality: Improving bionic vision with robotic sensing. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 304–307. IEEE, 2012.

[FUZOU-2015] Fuzou looklens optics co. ltd. http://www.looklens.com/camera-lens-tips/what-do-aspheric-or-aspherical-mean-3.html.

[GENG-2011] Jason Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011.

[GOOGLE]Google. Cardboard device. https://www.google.com/get/cardboard/.

[GUTIÉRREZ-GÓMEZ-2012] Daniel Gutiérrez-Gómez, Luis Puig, and José Jesús Guerrero. Full scaled 3d visual odometry from a single wearable omnidirectional camera. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4276–4281. IEEE, 2012.

[HIRSH-1989] Joy Hirsch and Christine A Curcio. The spatial resolution capacity of human foveal retina. *Vision research*, 29(9):1095–1101, 1989.

[HOLOLENS] Microsoft Corporation. Hololens device. https://www.microsoft.com/microsoft-hololens/en-us.

[HTC] HTC Corporation. Htc vive. http://www.htcvr.com/.

[IAPB-2010] International Agency for the Prevention of Blindness. 2010 report.

[IFIXIT-2015] Ifixit. Oculus rift development kit 2 teardown. https://es.ifixit.com/Teardown/Oculus+Rift+Development+Kit+2+Teardown/27613.

[INE] Instituto nacional de estadística.

[INSIDER-2015] Business Insider. Microsoft acquires canesta for patents. http://www.businessinsider.com/if-microsoft-acquires-canesta-its-probably-a-patent-play-2010-10.

[KINECT] https://www.microsoft.com/en-us/download/details.aspx?id=44561.

[LACHAT-2015] E Lachat, H Macher, MA Mittet, T Landes, and P Grussenmeyer. First experiences with kinect v2 sensor for close range 3d modelling. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS),* 2015.

[LI-2013] B. Li, A. Mian, W. Liu, A. Krishna. Using Kinect for Face Recognition Under Varying Poses, Expressions, Illumination and Disguise. *Applications of Computer Vision (WACV)*, 2013 *IEEE Workshop on*.

[LI-2014] Larry Li. Time-of-flight camera–an introduction.

[LÓPEZ-NICOLÁS-2014] G López-Nicolás, J Omedes, and JJ Guerrero. Spatial layout recovery from a single omnidirectional image and its matching-free sequential propagation. *Robotics and Autonomous Systems*, 62(9):1271–1281, 2014.

[MICROSOFT-KINECT] Microsoft. Kinect for xbox 360. http://www.xbox.com/en-US/xbox-360/accessories/kinect.

[OCULUS-2015] Oculus VR. Oculus rift development kit 2. https://www.oculus.com/en-us/dk2/.

[PRIMESENSE]PrimeSense. Apple primesense carmine 1.09 - 3d webcam sensor. http://www.amazon.com/dp/B00KO908MM?tag=tellgadget-20.

[PUIG-2014] Luis Puig, Jose J Guerrero, and Kostas Daniilidis. Scale space for camera invariant features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(9):1832–1846, 2014.

[SIRIGNANO-2015] Carmine Sirignano. Kinect v2 calibration parameters. https://social.msdn.microsoft.com/Forums/es-ES/c1d41e88-3539-4374-a5a0-0da6ce2cb877/kinect-v2-calibration-parameters-intrinsic-and-distortion-and-coordinatemapper-processing?forum=kinectv2sdk.

[SONY] Sony. Playstation vr. https://www.playstation.com/es-es/explore/ps4/features/playstation-vr/.

[SPECSMAG-2015] SpecsMag. Samsung galaxy note 3 specs. http://info.specsmag.com/gadget/4170.

[VIOLA-2004] P. Viola, M. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision* 57(2), 137–154, 2004.

[WANG-2015] Q. Wang, G. Kurillo, F. Ofli, R. Bajcsy. Evaluation of Pose Tracking Accuracy in the First and Second Generations of Microsoft Kinect. *arXiv:1512.04134v1 [cs.CV]* 13 Dec 2015.

[WEILAND-2014] James D Weiland and Mark S Humayun. Retinal prosthesis. *Biomedical Engineering, IEEE Transactions on*, 61(5):1412–1424, 2014.

[ZHANG-2010] C. Zhang, Z. Zhang. A Survey of Recent Advanes in Face Detection, 2010.