

UNIVERSIDAD CARLOS III DE MADRID
DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA



DISEÑO E IMPLEMENTACIÓN DE
ALGORITMOS DE TRATAMIENTO
DE IMÁGENES PARA AYUDAS
TÉCNICAS VISUALES USANDO
HMDS

PROYECTO FIN DE CARRERA
INGENIERÍA INDUSTRIAL

AUTOR: CARLOS A. BARRANCO GARCÍA

TUTOR: JUAN CARLOS TORRES ZAFRA



Universidad
Carlos III de Madrid

Departamento de Tecnología Electrónica

PROYECTO FIN DE CARRERA

DISEÑO E IMPLEMENTACIÓN DE
ALGORITMOS DE TRATAMIENTO DE
IMÁGENES PARA AYUDAS TÉCNICAS
VISUALES USANDO HMDS

Autor: Carlos A. Barranco García

Tutor: Juan Carlos Torres Zafra

Leganés, 14 abril de 2011

Título: Diseño e implementación de algoritmos de tratamiento de imágenes para ayudas técnicas visuales usando HMDs

Autor: Carlos A. Barranco García.

Tutor: Juan Carlos Torres Zafra.

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

AGRADECIMIENTOS

Es realmente complicado nombrar a todas las personas que en alguna ocasión me han prestado su apoyo, por lo que espero que no se me olvide nadie.

En primer lugar quiero dar las gracias a mi familia por ayudarme en todo momento, por haber estado a mi lado y haberme comprendido y escuchado cuando los he necesitado.

Agradezco también a mi novia y también compañera de estudios durante muchos años, por su comprensión y ayuda incondicional durante todo este tiempo, lo cual ha sido fundamental para que haya podido llegar hasta aquí.

Gracias al departamento de Tecnología Electrónica de la Universidad Carlos III de Madrid, por haberme dado la posibilidad de realizar este proyecto y por toda la ayuda que me han ofrecido. En especial quiero agradecersele a Juan Carlos Torres Zafra y a Ricardo Vergaz Benito los cuales me han ayudado y enseñado mucho tanto en este proyecto como en diferentes asignaturas de la titulación.

También quería agradecer a mis compañeros de universidad con los cuales he compartido muchas horas de clase. De entre los cuales, tengo que mencionar especialmente a Eva y Manu, con los cuales he pasado muchas horas de estrés en las prácticas de laboratorio.

En definitiva, muchas gracias a toda la gente, que de un modo u otro, me ha apoyado en algún momento de mi vida.

RESUMEN

Este proyecto se basa en el desarrollo de una herramienta destinada a la ayuda asistencial para personas con problemas derivados de la baja visión. En concreto, está enfocado para aportar una solución a personas con carencia de la percepción de distancias o profundidades.

Actualmente, las ayudas técnicas existentes no ofrecen soluciones para este tipo de discapacidad visual. Así, con este proyecto se intenta desarrollar e implementar un sistema que proporcione información visual referente a las distancias de los objetos que se encuentran en el entorno de las personas que requieren de esta ayuda.

La solución desarrollada está basada en el concepto de realidad aumentada y el tratamiento de imágenes. De esta manera, la información mostrada al usuario está compuesta de dos imágenes, una referente al entorno real y otra con información de las distancias. Esta última, es el resultado del cálculo de los contornos de los objetos presentes, pero representados con diferentes grosores y colores en función de la distancia.

Palabras clave: baja visión, correlación, disparidad, HMD, mapa de profundidad, realidad virtual, visión binocular, visión estereoscópica.

ABSTRACT

This project is based on developing a tool to help care for people with problems of low vision. In particular, it is focused on providing a solution to people who suffer a lack of vision over distance or with depth.

Currently, there are no technical aids to offer solutions for this type of visual impairment. As such, this project aims to develop and implement a system that provides visual information concerning the distances of objects found in the environment of people who require this assistance.

The solution developed is based on the concept of augmented reality and image processing. In this concept, the information displayed to the user consists of two images, one concerning the actual environment, and other information on the distances. The latter is the result of the calculation of the contours of the objects, but is represented with different thicknesses and colors depending on the distance.

Keywords: low vision, correlation, disparity, HMD, depth map, virtual reality, binocular vision, stereoscopic vision.

INDICE DE CONTENIDO

CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS	1
1.1. Introducción	1
1.2. Baja visión	2
1.3. Principales enfermedades de la visión.....	2
1.4. Ayudas técnicas	4
1.5. Visión binocular	7
1.6. Objetivos	8
1.7. Organización de la memoria.....	9
CAPÍTULO 2. PRESENTACIÓN DE SISTEMA.....	11
2.1. Descripción del sistema final.....	11
2.2. Descripción de los bloques.....	12
CAPÍTULO 3. SISTEMA DE SALIDA	13
3.1. Introducción	13
3.2. Realidad virtual y realidad aumentada	13
3.2.1. Definición	13
3.2.2. Aplicaciones	14
3.3. Gafas y cascos de realidad aumentada (HMD)	14
3.3.1. Definición y características.....	14
3.3.2. Sistemas ópticos frente a visuales	16
3.3.3. Enfoque y contraste.....	17
3.3.4. Comparación con realidad virtual	17
3.3.5. Problema de registro	18
3.4. Pantallas.....	18
3.5. Equipo utilizado	19
CAPÍTULO 4. SISTEMA DE ADQUISICIÓN DE IMÁGENES.....	21
4.1. Introducción	21
4.2. Cámaras	21
4.2.1. Etapa de pruebas	22
4.2.1.1. Cámara Creative	22
4.2.1.2. Cámara Minoru 3D	24
4.2.1.3. Otras opciones.....	25
4.2.2. Equipo utilizado	26

CAPÍTULO 5. SISTEMA DE PROCESAMIENTO	27
5.1. Introducción	27
5.2. Equipos susceptibles de ser utilizados como sistema de procesamiento	28
5.3. Equipo utilizado	29
5.4. Montaje final del sistema.....	30
CAPÍTULO 6. ALGORITMOS	31
6.1. Lenguaje de programación.....	31
6.2. Diseño de los algoritmos.....	32
6.2.1. Interfaz gráfica.....	32
6.2.2. Calibración mediante FFT.....	33
6.2.3. Adquisición de imágenes.....	35
6.2.4. Preprocesamiento	36
6.2.5. Cálculo de la distancia	37
6.2.5.1. Métodos de correlación	39
6.2.5.2. Cálculo de la correlación	40
6.2.5.3. Tiempo de cómputo	41
6.2.6. Cálculo de contornos	44
6.2.6.1. Cálculo contornos usando mapa de disparidad	46
6.2.6.2. Cálculo de contornos usando imagen original	48
CAPÍTULO 7. CONCLUSIONES.....	49
7.1. Resultados y conclusiones.....	49
7.2. Trabajos futuros.....	56
CAPÍTULO 8. PRESUPUESTO	57
REFERENCIAS	59
AXEXO 1. ALGORITMOS DESARROLLADOS.....	61
ANEXO 2. HOJAS DE CARÁCTERÍSTICAS.....	73
GLOSARIO	113

INDICE DE FIGURAS

CAPÍTULO 1

Figura 1.1. Visión con miopía.	3
Figura 1.2. Visión con astigmatismo.....	3
Figura 1.3. Visión con glaucoma.....	4
Figura 1.4. Lupa manual.	5
Figura 1.5. Telescopios montados en gafas.	6
Figura 1.6. Diagrama de visión binocular.	7

CAPÍTULO 2:

Figura 2.1. Diagrama de bloques del sistema final.	11
---	----

CAPÍTULO 3:

Figura 3.1. Head Mounted Display.....	14
Figura 3.2. HMD Trivisio ARVision-3D.	19

CAPÍTULO 4:

Figura 4.1. Cámara web Creative Live! Cam VF0470.	22
Figura 4.2. Sistema empleado para calibrar las cámaras Creative.....	23
Figura 4.3. Imagen tomada por cámara Creative.....	23
Figura 4.4. Cámara Minoru 3D.	24
Figura 4.5. Cámaras Zivora 3D (izquierda) y Thanko Webcam 3D (derecha).....	25
Figura 4.6. Fujifilm Finepix Real 3D W3 (izquierda) y Inlife Easy 3D (derecha).....	25
Figura 4.7. HMD Trivisio ARVision 3D.	26

CAPÍTULO 5:

Figura 5.1. Gráfica comparativa tiempos de cómputo entre Intel i7 y ATOM N270.	28
Figura 5.2. Conectores del HMD ARVision 3D.....	29
Figura 5.3. Adaptador HDMI a DVI.	29
Figura 5.4. Ordenador HP Envy 14-1180es.	29

CAPÍTULO 6:

Figura 6.1. Diagrama general de los algoritmos desarrollados.....	32
Figura 6.2. Interfaz gráfica desarrollada.	32
Figura 6.3. Ejemplo de búsqueda de un patrón usando FFT.....	33
Figura 6.4. Ejemplo búsqueda de un patrón usando FFT.....	34

Figura 6.5. Diagrama de flujo de la calibración.....	34
Figura 6.6. Proceso de calibración con cámara Minoru 3D.	35
Figura 6.7. Mapa de disparidad tras aplicación de diferentes filtros.....	36
Figura 6.8. Resultado final tras aplicación de diferentes filtros.....	37
Figura 6.9. Diagrama de flujo del algoritmo para el cálculo de la distancia.	37
Figura 6.10. Mapa de profundidad a color.....	38
Figura 6.11. Mapa de profundidad en escala de grises.	38
Figura 6.12. Para de imágenes estéreo Tsukuba[26].	38
Figura 6.13. Representación del área total en el que se calcula correlación.....	40
Figura 6.14. Fase de búsqueda de un pixel homólogo.....	41
Figura 6.15. Tiempos de cómputo de la correlación en imagen 64x64.	42
Figura 6.16. Tiempos de cómputo de la correlación en imagen 128x128.	42
Figura 6.17. Tiempos de cómputo de la correlación en imagen 256x256.	42
Figura 6.18. Tiempos de cómputo de la correlación en imagen 80x105.	42
Figura 6.19. Tiempos de cómputo de la correlación en imagen 120x160.	43
Figura 6.20. Tiempos de cómputo de la correlación en imagen 144x192.	43
Figura 6.21. Tiempos de cómputo de la correlación en imagen 240x360.	43
Figura 6.22. Representación de contornos a color, distancia 1.	45
Figura 6.23. Representación de contornos a color, distancia 2.	45
Figura 6.24. Representación contornos a color con diferentes grosores.	45
Figura 6.25. Máscara empleada en dilatación del contorno más próximo.....	45
Figura 6.26. Máscara para dilatación del segundo contorno.....	46
Figura 6.27. Máscara para dilatación del tercer contorno.....	46
Figura 6.28. Máscara para dilatación cuarto y quinto contorno.....	46
Figura 6.29. Diagrama de flujo del cálculo de contornos usando mapa de disparidad.	46
Figura 6.30. Mapa de profundidad de una imagen tomada con Minoru 3D.	46
Figura 6.31. Descomposición del mapa de profundidad en función del color y cálculo de los contornos.	47
Figura 6.32. Diagrama de flujo del cálculo de contornos usando imagen original.	48
Figura 6.33. Ejemplos del cálculo de contornos usando imagen original.	48

CAPÍTULO 7:

Figura 7.1. Resultado para barrido 10 y vecindad 3x3.....	49
Figura 7.2. Resultado para barrido 10 y vecindad 5x5.....	49
Figura 7.3. Resultado para barrido 10 y vecindad 7x7.....	50
Figura 7.4. Resultado para barrido 13 y vecindad 3x3.....	50
Figura 7.5. Resultado para barrido 13 y vecindad 5x5.....	50
Figura 7.6. Resultado para barrido 13 y vecindad 7x7.....	50
Figura 7.7. Resultado para barrido 15 y vecindad 3x3.....	51
Figura 7.8. Resultado para barrido 15 y vecindad 5x5.....	51
Figura 7.9. Resultado para barrido 15 y vecindad 7x7.....	51
Figura 7.10. Resultado para barrido 20 y vecindad 3x3.....	51
Figura 7.11. Resultado para barrido 20 y vecindad 5x5.....	52

Figura 7.12. Resultado para barrido 20 y vecindad 7x7.....	52
Figura 7.13. Resultado para barrido 25 y vecindad 3x3.....	52
Figura 7.14. Resultado para barrido 25 y vecindad 5x5.....	52
Figura 7.15. Resultado para barrido 25 y vecindad 7x7.....	53
Figura 7.16. Resultado para barrido 30 y vecindad 3x3.....	53
Figura 7.17. Resultado para barrido 30 y vecindad 5x5.....	53
Figura 7.18. Resultado para barrido 30 y vecindad 7x7.....	53
Figura 7.19. Tiempo cómputo para imágenes de 80x105.....	54
Figura 7.20. Tiempo de cómputo para imágenes de 120x160.....	54
Figura 7.21. Tiempo de cómputo para imágenes de 144x192.....	55
Figura 7.22. Tiempos de cómputo para imágenes de 240x320.	55

CAPÍTULO 8:

Figura 8.1. Diagrama de Gantt del proyecto.	57
--	----

CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS

1.1. Introducción

Este proyecto está destinado al desarrollo de una herramienta de ayuda para personas que carecen de percepción de la profundidad debido a problemas de baja visión. Antes de comenzar con el desarrollo de este proyecto, es necesario comentar la existencia de dos Proyectos Fin de Carrera realizados por el Grupo de Displays y Aplicaciones Fotónicas (GDAF) de la Universidad Carlos III de Madrid en colaboración con el Instituto de Oftalmobiología Aplicada (IOBA) de la Universidad de Valladolid cuyos objetivos siguen líneas comunes con este trabajo.

El primer trabajo se realizó en 2008 por Francisco Collado [1] y buscaba la adicción de los contornos de los objetos que conformaban la escena. Sin embargo, esta solución a pesar de diferenciar bien cada objeto no ofrece datos sobre la distancia a la que se encuentran.

Así, en el proyecto desarrollado en 2010 por Rafael Girón [2], se abordó el problema de los sistemas de ayuda existentes que empleaban una única cámara, por lo que los pacientes sufrían la pérdida de la percepción de la profundidad y distancia de los objetos. En el trabajo mencionado se realiza un estudio sobre diferentes métodos para el cálculo de las distancias de objetos haciendo uso de la visión estereoscópica. En este trabajo se muestra al usuario un mapa de profundidad de la escena, lo cual dificulta asimilar la información por parte del paciente, al no mostrarse la escena real.

Tras valorar los pros y los contras de los proyectos anteriores, con este proyecto, se decidió abordar una solución basada en los aspectos positivos de cada uno de estos trabajos.

Para una mejor comprensión del trabajo, a continuación se da una breve descripción del concepto de baja visión, principales patologías que la provocan, ayudas técnicas existentes, visión binocular o estereoscópica y finalmente se describe el objetivo final buscado con el desarrollo de este proyecto.

1.2. Baja visión

La baja visión es una visión insuficiente que aun con gafas, lentes correctivas, tratamiento médico o quirúrgico, no puede ser corregida para poder realizar una tarea deseada. Desde el punto de vista funcional, pueden considerarse como personas con baja visión aquellas que poseen un resto visual suficiente para ver la luz, orientarse por ella y emplearla con propósitos funcionales.

La Organización Mundial de la Salud (OMS) definió en 1992 una persona con baja visión a aquella con una incapacidad en la función visual aun después de tratamiento, con una agudeza visual en el mejor ojo menor de 0.3 a percepción de luz o un campo visual inferior a 10° desde el punto central de fijación, pero suficiente visión para ser utilizada en la planificación o ejecución de una tarea.

En el mundo hay 314 millones de personas con discapacidad visual, 45 millones de las cuales padecen ceguera. La mayoría de las personas con discapacidad visual tienen edad avanzada, y el riesgo es mayor para las mujeres a todas las edades y en todo el mundo. Aproximadamente el 87% de las personas con discapacidad visual en el mundo viven en países en desarrollo y el 82% son mayores de 50 años, pese a que solo representan un 19% de la población mundial.

Aproximadamente un 85% de los casos mundiales de discapacidad visual son evitables y la corrección de los errores de refracción podría devolver una visión normal a más de 12 millones de niños de 5 a 15 años. [3]

Las tendencias mundiales muestran desde principios de los años noventa una disminución de las tasas mundiales de discapacidad visual y variaciones en la distribución de sus causas. La discapacidad visual y la ceguera debidas a enfermedades infecciosas han disminuido mucho, pero hay un aumento notorio del número de personas con discapacidad visual o ceguera debido a trastornos relacionados con el aumento de la esperanza de vida. [4] [5] [6]

1.3. Principales enfermedades de la visión

La baja visión puede estar causada por determinadas patologías asociadas muchas veces a la edad, traumatismos o enfermedades congénitas. Los problemas más frecuentes son los que tienen que ver con la refracción, pero también puede tratarse de problemas crónicos o funcionales que no permiten ver adecuadamente.[7]

Los principales problemas de refracción son la miopía, hipermetropía y astigmatismo, los cuales se describen a continuación.

- **Miopía.**

Se presenta cuando los rayos de luz se enfocan delante de la retina y no directamente sobre ella haciendo que una persona con miopía tenga dificultades para enfocar bien los objetos lejanos. Este problema se debe a una deformación del globo ocular, que se presenta alargado, o porque el cristalino tiene una distancia focal menor a la normal. La miopía se corrige con lentes divergentes o en algunos casos puede recurrirse a la cirugía.



Figura 1.1. Visión con miopía.

- **Hipermetropía.**

Al contrario que en la miopía los rayos de luz procedentes del infinito se enfocan en un punto situado detrás de la retina, la consecuencia es que la imagen es borrosa y puede existir por lo tanto una falta de agudeza visual. Es un defecto que se trata mediante el uso de lentes correctoras.

- **Astigmatismo.**

Estado ocular que generalmente proviene de un problema en la curvatura de la córnea, lo que impide el enfoque claro de los objetos tanto lejos como cerca. La córnea, que es una superficie esférica, sufre un achatamiento en sus polos, lo cual produce distintos radios de curvatura en el eje del ojo, por ende cuando la luz llega al ojo, específicamente en la córnea, la imagen que se obtiene es poco nítida y distorsionada. Se corrige con el uso de lentes tóricas o mediante soluciones quirúrgicas, como la cirugía refractiva.

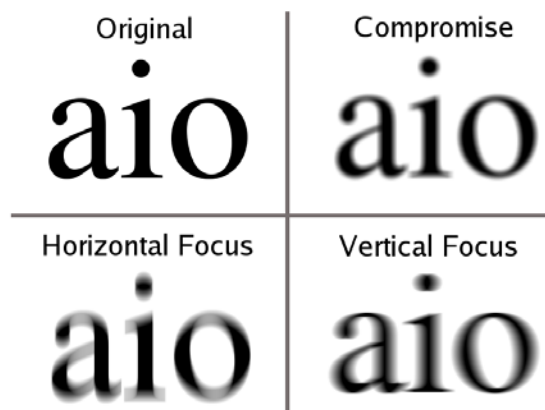


Figura 1.2. Visión con astigmatismo.

Los principales problemas de visión crónicos o funcionales son:

- **DMAE (Degeneración Macular Asociada a la Edad).**

Es una enfermedad del ojo ocasionada por degeneración, daños o deterioro de la mácula. La mácula es una capa amarillenta de tejido sensible a la luz que se encuentra en la parte posterior del ojo, en el centro de la retina, esta área proporciona la agudeza visual que permite al ojo percibir detalles finos y pequeños. Cuando la mácula no funciona correctamente, las áreas del centro del campo visual empiezan a perder nitidez, se produce pérdida de la visión central, distorsión de líneas rectas, problemas para detallar y áreas oscuras.

- **Glaucoma.**

Enfermedad en la que se ve afectado el nervio óptico, principalmente inducida por presión intraocular alta. Una consecuencia habitual es la pérdida del campo de visión y puede degenerar en la ceguera total del ojo afectado.



Figura 1.3. Visión con glaucoma.

- **Cataratas.**

Es la opacificación parcial o total del cristalino que produce pérdida de visión. El tratamiento de las cataratas es principalmente quirúrgico. Siendo la principal causa de ceguera en todo el mundo, excepto en los países más desarrollados. [4] [5] [6][10] [11][12]

1.4. Ayudas técnicas

Las ayudas técnicas de baja visión son aparatos técnicos específicos que mejoran la visión, aprovechando el resto de visión útil. Existen diferentes ayudas técnicas para la visión, pero no existe ninguna que mejore la visión en cualquier circunstancia, siendo necesario diferentes ayudas en función del propósito.

Las ayudas técnicas pueden clasificarse en ayudas ópticas, no ópticas y electrónicas. Una **ayuda óptica** es un sistema óptico formado por lentes de elevada potencia que ayudan a personas con baja visión a aprovechar al máximo su resto visual. Estas ayudas no deben confundirse con las gafas convencionales. De entre estas ayudas se pueden destacar:

- **Lupa.**

La lupa es la ayuda de baja visión más conocida y de mayor facilidad de manejo. Forma parte del grupo de ayudas ópticas recomendadas para tareas de cerca. Permite una mayor distancia de trabajo que un microscopio pero un campo visual menor. Se fabrican a partir de lentes positivas (que producen aumento de la imagen) y pueden usarse de forma manual o llevar un soporte para apoyar en el papel. Las lupas pueden ser con soporte, de mano, iluminadas y no iluminadas.



Figura 1.4. Lupa manual.

El enfoque, es necesario para un rendimiento óptimo de las lupas, en el caso de lupas manuales deberá realizarse alejando y acercando la lupa al texto.

- **Gafas de alto poder dióptrico.**

Lentes también positivas que se incorporan en una montura y que se pueden utilizar para lecturas prolongadas, tareas de punto, costura, escritura, etc.

También son denominadas microscopios. Estas gafas consiguen el aumento por disminución de la distancia, que se produce cuando reducimos la distancia del objeto al ojo. En realidad, no aumentan nada por sí mismos, pero facilitan la acomodación, que es la función visual que permite acercarse al objeto para verlo con nitidez.

Para algunas tareas, son más cómodos que las lupas porque permiten tener las manos libres.

Existen gafas monoculares y binoculares y pueden utilizarse de forma monofocal (con un solo foco), bifocal (con dos focos), trifocal (tres focos) o multifocal (varios focos).

La distancia de trabajo con estas lentes es corta, por lo que se requiere una correcta posición a la hora de leer para conseguir y mantener una imagen nítida.

- **Telescopios.**

Permiten aumentar el tamaño de los objetos que se encuentran a distancias lejanas. Pueden ser manuales o ir montados en gafas y a su vez pueden utilizarse con un solo ojo (monoculares) o con los dos (binoculares). Su poder de enfoque les hace útiles para todas aquellas tareas puntuales de visión lejana, como son la lectura de rótulos, teatros, TV, escaparates, etc. Este tipo de ayuda no permite su utilización en movimiento.

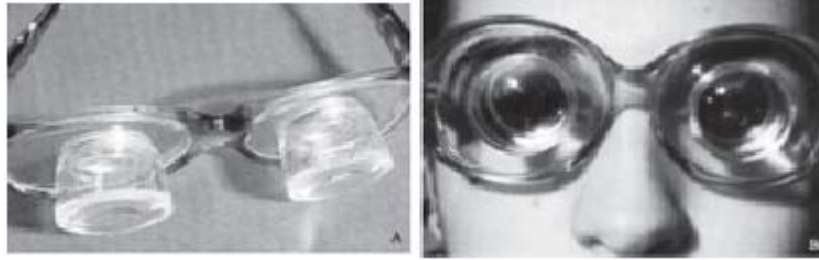


Figura 1.5. Telescopios montados en gafas.

Algunas **ayudas ópticas electrónicas** existentes son:

- **Lupas televisión o circuitos cerrados de televisión (CCTV)**

Ayudas que aumentan el tamaño de las imágenes y que constan de un monitor, una cámara y un sistema óptico. Cada usuario puede establecer la modalidad más acorde con sus condiciones visuales, en cuanto al tamaño, contraste y color necesario. Pueden utilizarse en sistemas blanco y negro y color, siendo muy variados los modelos existentes, algunos de ellos ofrecen la posibilidad de compatibilizar el monitor con un ordenador, permitiendo mediante la división de la pantalla, presentar las imágenes procedentes de la cámara y las del ordenador al mismo tiempo. Es un sistema de lectura muy apropiado para aquellos casos en los que la visión es muy reducida o el uso de las ayudas antes mencionadas resulte poco adecuado para conseguir sus objetivos.

Estas ayudas deben recomendarse desde un servicio especializado, que garantice los entrenamientos adecuados y los adapte a las necesidades individuales de cada persona con discapacidad visual.

Las **ayudas no ópticas** son aquellas que favorecen la utilización del resto visual o que potencian el rendimiento con las ayudas electrónicas y ópticas. Dentro de este grupo se incluyen ayudas:

- Para controlar los problemas de deslumbramiento: filtros con diferentes niveles de absorción.
- Para lectoescritura: guías para firmar, rotuladores, papeles pautados, atriles...
- Materiales para costura: enhebradores, cintas métricas con marcas, agujas de cabeza hendida.
- Relojes, despertadores y avisadores de tiempo adaptados en Braille, sonoros.
- Ayudas al desplazamiento, fundamentalmente bastones de movilidad, tarjeta "taxi", etc. [4] [6] [7] [8] [9][10] [11]

1.5. Visión binocular

El concepto de visión binocular es el aspecto de mayor importancia en el que se basa el desarrollo de este proyecto.

La visión binocular o visión estereoscópica es la capacidad que tienen ciertos seres vivos de integrar las dos imágenes capturadas por cada ojo en una sola por medio del cerebro (sistema nervioso central). Éste último, es el encargado de percibir las sensaciones que tanto un ojo como el otro están viendo y de enviar una respuesta única y en tres dimensiones.

Para tener una visión binocular es preciso que las órbitas oculares estén frontalizadas, puesto que de este modo el área de incidencia de la visión de ambos ojos es prácticamente idéntica, permitiendo una visión tridimensional de casi la totalidad del espacio visual. Es pues un tipo de visión que pierde amplitud de campo por ganar profundidad de campo.

La visión binocular ofrece a los seres humanos dos ventajas fundamentales:

- Aumento del campo visual
- Obtener una imagen virtual mejorada utilizando las dos imágenes que vienen de cada ojo. Esto es lo que se denomina visión binocular propiamente dicha.

Cada ojo por separado tiene un campo visual de unos 150° , pero los ojos no trabajan de forma independiente, sino que por el contrario, los dos ojos están enfocando al mismo punto. Eso significa que una buena parte del campo visual de un ojo se solapa con el otro. Es decir, un objeto que vemos con un ojo, si no está muy periférico, también lo vemos con el otro. Por tanto, el cerebro recibe información redundante y una parte grande del campo visual es percibida por los dos ojos. [4]

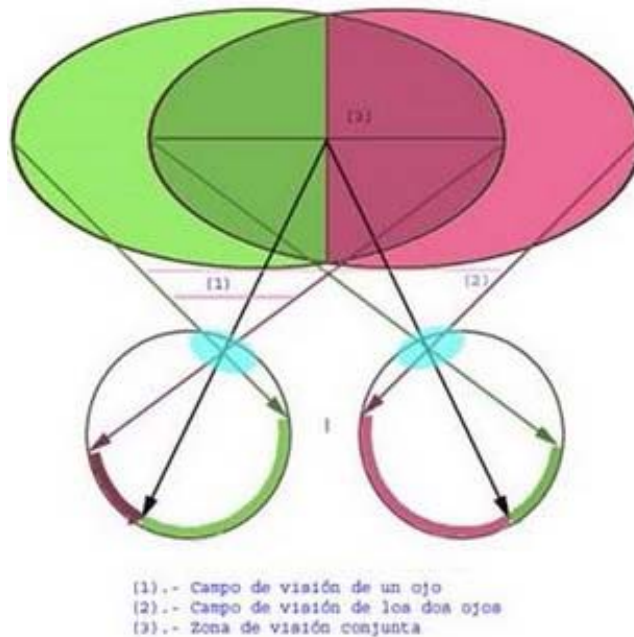


Figura 1.6. Diagrama de visión binocular.

Por lo tanto, si tenemos dos imágenes tomadas desde posiciones ligeramente diferentes y las mostramos por separado a cada ojo, el cerebro es capaz de reconstruir la distancia (y por lo tanto la profundidad) analizando la disparidad o el paralelismo entre estas imágenes, este proceso se llama estereopsis.

En el ser humano, la binocularidad es la situación normal, y exige de varias condiciones que se tienen que dar. En el caso de falla de alguna de ellas, el ser humano se ve privado de la visión binocular. Estas condiciones necesarias son que:

- Debe existir coordinación milimétrica entre un ojo y otro.
- El cerebro tiene que estar coordinado y disponer de la capacidad de análisis necesaria para procesar la información que recibe. Esta habilidad se desarrolla desde el momento del nacimiento.
- Es necesario una visión aceptable por los dos ojos. Una lesión o reducción en la capacidad de ambos o de uno de solo de los ojos puede reducir la calidad de la visión binocular.

Para tener una correcta visión binocular y llevar a cabo la estereopsis, es necesario que se realicen dos procesos:

- Fusión motora: Es la capacidad de enfocar exactamente al mismo sitio con los dos ojos.
- Función sensorial: El cerebro fusiona las dos imágenes captadas en una sola [12] [13].

1.6. Objetivos

El objetivo principal de este proyecto es mejorar la calidad de vida de personas con baja visión, permitiéndoles recuperar la visión binocular de las que se han visto privadas ya sea por una patología o disfunción visual y que por tanto no pueden distinguir con claridad las profundidades y distancias a las que se encuentran los objetos.

El objetivo de este proyecto es que un paciente con baja visión pueda diferenciar qué objetos del entorno en el cual se encuentra están cerca, a media distancia o lejos. Para ello se pretende desarrollar un sistema de ayuda óptica electrónica, que haciendo uso de gafas de realidad virtual y los algoritmos desarrollados en este trabajo, permita ver el entorno real, pero con la peculiaridad de que los objetos contengan información sobre su distancia, al ser mostrados al paciente bordeados de un color u otro en función de ésta.

1.7. Organización de la memoria

El primer capítulo está orientado a dar una introducción, con la cual informar al lector sobre el problema que supone la baja visión para las personas que la padecen, las patologías que pueden provocarla y las diferentes ayudas técnicas existentes. Además se ha informado sobre las motivaciones y objetivos del proyecto para lo cual se ha explicado la importancia de la visión binocular en la percepción de distancias y profundidades.

En el segundo capítulo se muestra esquemáticamente el sistema final a implementar y se describe brevemente la función que realiza cada bloque del sistema.

El tercer capítulo está destinado para presentar y describir el sistema encargado de mostrar los resultados al usuario. Se explican los conceptos de realidad virtual y realidad aumentada, así como los sistemas de salida necesarios para llevar a cabo aplicaciones de tecnología asistencial. Finalmente, se presenta el dispositivo utilizado.

El cuarto capítulo se centra en el sistema de adquisición de imágenes, donde se presentan los sistemas empleados en las fases iniciales del desarrollo de este proyecto y el seleccionado para la implementación del sistema final. Además se incluyen una lista de cámaras comerciales que podrían ser empleadas como alternativa de sistema de adquisición de imágenes.

El quinto capítulo está dedicado al sistema de procesamiento. En el cual se explica la función y requisitos mínimos que ha de tener este sistema. Finalmente se presenta el sistema de procesamiento elegido y se da una visión general del montaje final de todos los dispositivos que conforman el conjunto del sistema final.

En el sexto capítulo inicialmente se da una breve introducción sobre la herramienta software que se ha empleado para el desarrollo de los algoritmos. Posteriormente se describen los algoritmos desarrollados y necesarios para alcanzar el objetivo de este trabajo, resumiéndose en tablas y gráficas los tiempos de cómputo en función del método y resolución de cálculo.

En el séptimo capítulo se valoran los resultados obtenidos y se proponen una serie de ideas para mejorar la herramienta desarrollada.

En el octavo capítulo inicialmente se muestra un diagrama de Gantt que muestra las fases que ha seguido el proyecto y finalmente se muestra el presupuesto del proyecto realizado.

Una vez descrito el trabajo realizado, se indican las referencias de interés que han sido de utilidad para el desarrollo de éste.

El anexo 1 está dedicado a dar a conocer al lector los algoritmos que se han desarrollado e implementado.

El anexo 2 muestra las hojas de características de los equipos empleados en el desarrollo de este trabajo.

Finalmente se muestra un glosario con los términos usados en esta memoria.

CAPÍTULO 2. PRESENTACIÓN DE SISTEMA

2.1. Descripción del sistema final

Tras haber descrito brevemente el problema que se pretende abordar, se procederá en este capítulo a presentar de forma resumida el sistema final a implementar y con el que se pretende alcanzar los objetivos marcados.

El sistema final del proyecto consta de cinco bloques (Figura 2.1) claramente diferenciados, los cuales cumplen una función concreta.

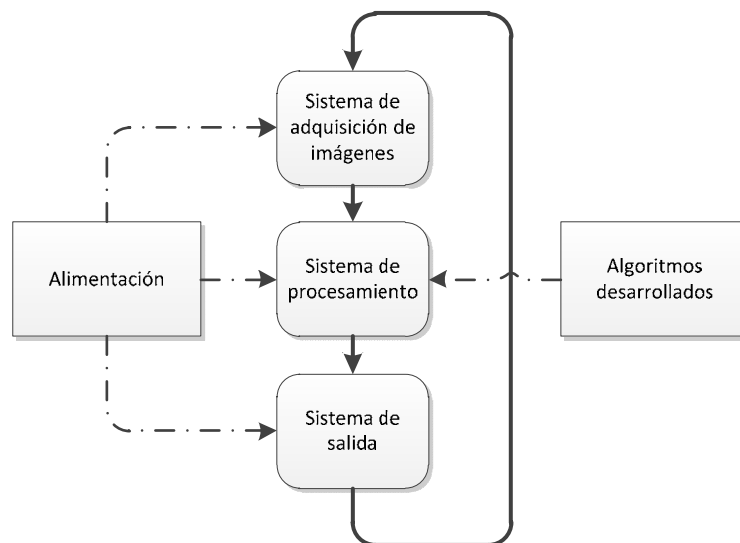


Figura 2.1. Diagrama de bloques del sistema final.

2.2. Descripción de los bloques

Los bloques mostrados en la figura anterior, se describen a continuación:

- **Sistema de adquisición de imágenes.**

Este sistema es el encargado de capturar la información necesaria del entorno para posteriormente procesarla y mostrársela al usuario. El sistema encargado de obtener esta información está compuesto de dos cámaras con similares características y que deben cumplir unos requisitos en cuanto a su ubicación y calidad.

La ubicación de este sistema es de vital importancia, para ofrecer lo más fielmente posible la visión del entorno al usuario. Para ello, es importante que las cámaras estén situadas a la altura de los ojos y separadas una distancia similar a la existente entre los ojos humanos.

- **Sistema de salida.**

Es el encargado de mostrar los resultados obtenidos tras el procesamiento de las imágenes capturadas por el sistema anterior. Este sistema constará de dos pantallas, en las cuales se mostrarán dichos resultados. El sistema de salida debe ser ergonómico para facilitar al usuario su utilización.

- **Sistema de procesamiento.**

Este dispositivo supone un nexo de unión entre los dos sistemas anteriores, ya que es el encargado de procesar las imágenes captadas por el sistema de adquisición y enviar los resultados obtenidos tras el procesamiento al sistema de salida para ser mostrados al usuario.

El sistema de procesamiento tendrá unas restricciones en cuanto a portabilidad, autonomía, velocidad de procesamiento y conexiones. No sería útil un sistema cuya velocidad de procesamiento no mostrase los resultados en tiempo real, ni un sistema que debido a su peso o tipo de alimentación no permitiese la movilidad del usuario. Además, este sistema es necesario que disponga de las conexiones necesarias para comunicarse con los otros sistemas.

- **Algoritmos.**

Las imágenes son procesadas haciendo uso de los algoritmos desarrollados e implementados. Siendo éste el bloque del proyecto que más esfuerzo ha requerido. La realización de los algoritmos ha requerido un estudio previo sobre cálculo de profundidades y visión estereoscópica. Los algoritmos deben cumplir con una serie de restricciones en cuanto a tiempos de cómputo, con objeto de mantener la sensación de continuidad buscada en la presentación de resultados al usuario.

- **Alimentación.**

Como se mencionó, este proyecto busca ofrecer una ayuda electrónica a personas con baja visión, por lo que el sistema estará compuesto por una serie de dispositivos electrónicos. Cada uno de estos dispositivos requiere de un sistema alimentación para su funcionamiento, el cual debe ser portable, por lo que la autonomía del sistema dependerá de este sistema y del consumo de los dispositivos electrónicos empleados.

CAPÍTULO 3. SISTEMA DE SALIDA

3.1. Introducción

El sistema de salida es el encargado de mostrar las imágenes que fueron capturas por el sistema de adquisición una vez procesadas, haciendo uso del sistema de procesamiento y de los algoritmos desarrollados, los cuales son explicados en los siguientes capítulos.

Para ello, se ha decido emplear como sistema de salida unas gafas de realidad virtual, debido a su ergonomía, diseño y portabilidad.

3.2. Realidad virtual y realidad aumentada

3.2.1. Definición

La realidad virtual es un sistema tecnológico, basado en el empleo de ordenadores y otros dispositivos, cuyo fin es producir una apariencia de realidad que permita al usuario tener la sensación de estar presente en ella. Se consigue mediante la generación por ordenador de un conjunto de imágenes, las cuales son contempladas por el usuario a través de un casco provisto de un visor especial. Su aplicación, inicialmente estaba centrada en el terreno de los videojuegos, pero se ha extendido a otros muchos campos, como la medicina o simuladores de vuelo.

El término de realidad aumentada se usa para definir una visión directa o indirecta de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta a tiempo real. Consiste en un conjunto de dispositivos que añaden información virtual a la información física ya existente, es decir, añadir una parte sintética virtual a lo real. Esta es la principal diferencia con la realidad virtual, puesto que no sustituye la realidad física, sino que sobrepone los datos informáticos al mundo real.

Los sistemas de realidad aumentada son definidos en la manera de que requieren cascos de realidad virtual o *Head-Mounted Displays* (HMDs), sobre los cuales se hace un estudio más detallado en el apartado 3.3.

3.2.2. Aplicaciones

La realidad aumentada ofrece infinidad de nuevas posibilidades de interacción, que hacen que esté presente en muchos y varios ámbitos, como son la arquitectura, entretenimiento, educación, arte, medicina o comunidades virtuales. [4] [14]

La mayoría de aplicaciones de realidad aumentada para proyectos educativos se usan en museos, exhibiciones, parques de atracciones temáticos, etc., ya que su coste no es lo suficientemente bajo como para ser usado en el ámbito doméstico. Un ejemplo en el ámbito educativo, es la editorial RaRaRa Editores cuyo objetivo es aumentar el interés por el aprendizaje mediante el uso de libros educativos con realidad aumentada.

En el ámbito de la medicina, la realidad aumentada es usada principalmente en el campo de la cirugía. Así, se permite al cirujano superponer datos visuales como por ejemplo termografías o la delimitación de los bordes limpios de un tumor, invisibles a simple vista, minimizando el impacto de la cirugía.

La realidad aumentada puede aumentar la efectividad de sistemas de navegación. Por ejemplo, mediante uso de *displays* en los cristales de los automóviles, puede aportarse al conductor información sobre la navegación e incidencias del tráfico.

Aparte de las aplicaciones mencionadas anteriormente, la realidad aumentada es de gran utilidad en muchas más aplicaciones, como puede ser la reconstrucción virtual de edificios históricos destruidos.

3.3. Gafas y cascos de realidad aumentada (HMD)

3.3.1. Definición y características

Los cascos de realidad aumentada o más conocidos como *Head-Mounted Display* (HMD) son dispositivos de visualización similar a un casco, que permiten reproducir imágenes creadas por ordenador sobre un *display* muy cercano a los ojos o directamente sobre la retina de los ojos. En este segundo caso el HMD recibe el nombre de monitor virtual de retina.

Los HMDs proveen al usuario de un conjunto de capacidades que un *display* convencional no puede conseguir.



Figura 3.1. Head Mounted Display

Los HMD pueden ser de dos tipos:

- Monocular: las imágenes creadas por ordenador sólo se reproducen sobre un ojo, mientras que el otro ojo continúa mirando al entorno real del usuario. Estos equipos ofrecen una mayor ergonomía y comodidad.
- Binocular: las imágenes creadas por ordenador se reproducen sobre los dos ojos, obteniendo así una imagen estereoscópica. Cada pantalla puede tener ópticas separadas, pero la imagen es esencialmente idéntica para ambos ojos.

Debido a su proximidad con los ojos, el *Head-mounted Display* consigue que las imágenes visualizadas resulten mucho mayores que las percibidas por pantallas normales, y permiten incluso englobar todo el campo de visión del usuario. Gracias a que el *display* se encuentra sujeto al casco, este puede seguir los movimientos del usuario, consiguiendo así que éste se sienta integrado en los ambientes creados por ordenador.

Algunos HMD reducen el campo de visión del usuario, de modo que el usuario no tiene influencias visibles del entorno que lo rodea, permitiendo así la completa inmersión de éste en una realidad virtual, ya que sólo percibirá las imágenes creadas por ordenador y que son reproducidas sobre el *display*. [4] [14][16][17][18]

Se pueden identificar cuatro áreas fundamentales que se deben satisfacer en el diseño o proceso de selección de HMDs:

- Requerimientos visuales: existen ciertos requerimientos en cuanto a contraste, luminancia, enfoque y alineamiento, color y resolución. Estos requerimientos son derivados de la comprensión de cómo la visión humana funciona y del conocimiento de las condiciones visuales del ambiente.
- Requerimientos físicos: temas críticos de la antropometría, centro de gravedad de la cabeza y el cuello, y la comodidad deben satisfacerse.
- Requerimientos del ambiente: las funciones de los seres humanos deben realizarse dentro de un estrecho rango de condiciones ambientales.
- Requerimientos del interfaz: cualquier control del HMD debe ser accesible, fácil de entender y fácil de usar.

Las gafas de realidad virtual o HMDs tienen una serie de características que los hacen más o menos óptimos para la realización de una función concreta. Estas características son enumeradas a continuación.

- Campo de visión: es la porción del espacio (medida en grados) que es capaz de captar.
- *Eye relief*: es la distancia desde la superficie de la horma de un ocular en la que el ojo puede obtener el ángulo de visión completo.
- Calidad de la imagen: define cómo de bien el usuario del HMD podrá interpretar con el *display*.
- Abertura del diafragma.
- Luminancia: se define como la densidad superficial del flujo luminoso.
- Enfoque: en los siguientes apartados se describe los problemas de enfoque que se producen al usar tecnologías ópticas y visuales. [17]

3.3.2. Sistemas ópticos frente a visuales

Un punto importante en el diseño de un sistema de realidad aumentada es como alcanzar la combinación entre lo real y lo virtual, existiendo dos tipos de configuraciones para alcanzar tal fin. Estas son las basadas en tecnologías ópticas y tecnologías de video. A continuación se realiza una comparación entre estos sistemas y se dan a conocer los puntos débiles y fuertes de cada uno.

Un sistema óptico o *see-through* HMD, es un dispositivo empleado para la combinación del mundo real y el virtual, que permite ver al usuario el mundo real pero con objetos virtuales superpuestos ya que poseen pantallas que permiten ver a través de ellas. En contraste, los sistemas *close-view* HMD no permiten la vista directa del mundo real.

Los sistemas *see-through* trabajan colocando combinadores ópticos delante del ojo del usuario. Estos son parcialmente transparentes, así el usuario puede ver directamente a través de ellos el mundo real.

Los combinadores ópticos normalmente reducen la cantidad de luz que el usuario ve del mundo real. Estos combinadores actúan como espejos semitransparentes, permitiendo la entrada de solo parte de la luz del mundo real. Así ellos pueden reflejar parte de la luz de los monitores sobre los ojos del usuario.

En contraste con los sistemas ópticos, los sistemas de video *see-through* HMDs funcionan combinando un *closed-view* HMD con una o dos video cámaras colocadas en el casco. Las videocámaras proveen al usuario de vistas del mundo real. Los videos de estas cámaras son combinados con las imágenes creadas.

En el caso de poseer de dos cámaras, se puede obtener información sobre la profundidad de cada pixel de las imágenes del mundo real. Esto permitiría a los objetos reales cubrir a los virtuales o viceversa.

Los sistemas ópticos o *see-through* tienen las siguientes ventajas sobre los de vídeo o *close-view*:

- Simplicidad: los sistemas de mezcla óptica son más simples y baratos que los de video. Esto es debido a que los sistemas ópticos solo tienen un flujo de video de que preocuparse, ya que la visión del mundo real se realiza de manera directa.
- Resolución: los sistemas de mezclado de video limitan que ve el usuario, tanto del mundo real como del virtual, a la resolución de la pantalla.
- Seguridad: en los sistemas de video *closed-view* HMDs si la alimentación es cortada por cualquier motivo, la persona queda completamente ciega, al no percibir ni el mundo real ni el virtual.
- No offset: los sistemas ópticos no tienen el problema de que la cámara este situada en un punto no correspondiente con la posición exacta de los ojos.

En cuanto a las ventajas que los sistemas de video ofrecen sobre los sistemas ópticos, podemos destacar:

- Flexibilidad en composición de estrategias.
- Amplio campo de visión.
- Retraso entre vista virtual y real pueden ser eliminados.
- Más facilidad para igualar los niveles de brillo de la imagen real y la virtual, al ser ambas imágenes generadas.

3.3.3. Enfoque y contraste

El enfoque puede ser un problema tanto para sistemas ópticos como para los de video. En los sistemas basados en video, tanto la imagen real como la virtual son proyectadas a la misma distancia. Sin embargo, dependiendo de la configuración del enfoque y campo de profundidad de la cámara, partes del mundo real pueden no estar enfocadas.

En el caso óptico, la imagen virtual es proyectada a una distancia desde el usuario. Esta distancia puede ser ajustable, aunque a veces es fija. Por consiguiente, mientras los objetos reales están a diferentes distancias del usuario, los objetos virtuales son todos proyectados a la misma distancia.

El contraste es otro problema, ya que la mayoría de los dispositivos no pueden alcanzar los mismos niveles de contraste que el ojo humano. Este problema no es tan severo en los sistemas de video, ya que tanto las vistas reales como las virtuales son generadas por los monitores. [15]

3.3.4. Comparación con realidad virtual

Los requerimientos generales de los sistemas de realidad aumentada pueden ser resumidos mediante la comparación con los requerimientos de los sistemas de realidad virtual, por tres subsistemas básicos que ambos requieren.

- Generador de la escena: los sistemas de realidad virtual requieren de mayores prestaciones ya que ellos reemplazan completamente el mundo real por un entorno virtual.
- Display: los *displays* usados en AR pueden tener requerimientos menores que los usados en VR, de nuevo debido a que en AR el mundo real no es reemplazado.
- Seguimiento y detección: en este caso los requerimientos para AR son más estrictos. La razón principal es el problema de registro, el cual se describe en el siguiente apartado. [15][17]

3.3.5. Problema de registro

Uno de los mayores problemas que actualmente limitan las aplicaciones de realidad aumentada es el problema de registro. Los objetos en el mundo real y virtual deben estar adecuadamente alineados uno respecto al otro.

Por ejemplo, un usuario que lleva un *closed-view* HMD puede ver en lugar de su mano real, una mano virtual que debería estar situada donde exactamente su mano real está, pero si la mano se encuentra ligeramente desplazada puede que el error sea inapreciable por el usuario. Sin embargo en un sistema *see-through* HMD, este error es más evidente, existiendo un conflicto visual-visual.

Además el problema conocido como captura visual hace incluso más difícil detectar estos errores. Este concepto se define como la tendencia del cerebro a creer lo que ve, en lugar de lo que percibe por otros sentidos, es decir, la información visual se impone sobre el resto de sentidos.

Así, se concluye que la realidad aumentada requiere de mayor precisión que la realidad virtual. [15]

3.4. Pantallas

En función de las pantallas empleadas en los HMD, se puede distinguir entre cascos con pantallas ópticas transparentes y los de pantallas no transparentes (estos sistemas fueron comparados en apartados anteriores).

En el sistema de la pantalla óptica, la instalación se compone de una serie de proyectores digitales, los cuales, simultáneamente muestran imágenes individuales sobre la pantalla transparente HOE. Estas son proyectadas desde diferentes ángulos. Cada una de estas imágenes es visible dentro de un pequeño ángulo de visión, y esto permite que cada ojo tenga una perspectiva diferente de dicha imagen proyectada. El HOE, es adecuado para las aplicaciones de realidad aumentada ya que proporciona unas cualidades perfectas de transparencia, además de poseer mucho brillo y por no obstruir la visión del mundo real que se ve a través de ella.

Las pantallas de los HMD pueden ser del tipo LED, OLED (*Organic Light Emitting Diode*), CRTs, LCoS (cristal líquido sobre silicio) o LCD (*Liquid Cristal Display*). Siendo las tecnologías más comunes a día de hoy, el desarrollo en dispositivos LCD y OLED.[14][15][18]

3.5. Equipo utilizado

El equipo final escogido para la presentación de los resultados en este proyecto ha sido impuesto por motivos de disponibilidad. El Grupo de Displays y Aplicaciones Fotónicas, GDAF, del Departamento de Tecnología Electrónica de la Universidad Carlos III de Madrid ha adquirido recientemente un equipo para la realización de diferentes proyectos entre los cuales se incluye éste. Se trata de un HMD de la marca Trivisio, en concreto se ha empleado el modelo ARvision-3D. En el anexo 2 se muestra el manual de instrucciones y hojas de características de este modelo.[18]



Figura 3.2. HMD Trivisio ARvision-3D.

Este HMD es de tipo binocular y su configuración es visual o *close view*. Las principales características de este dispositivo se muestran a continuación:

- Dimensiones (W/H/D): 150mm/42mm/84mm.
- Peso: 290g.
- Distancia entre ojos: 55-72mm ajustables.
- Displays: 2xSVGA AMLCD 800x600 color (equivalente a 1.440.000 pixels).
- Consumo de potencia: Display 2.5W (5V, tomado desde puerto USB).
- Interfaz de video: DVI-D.

CAPÍTULO 4. SISTEMA DE ADQUISICIÓN DE IMÁGENES

4.1. Introducción

Existen varios sistemas capaces de abordar el cálculo de distancias para simular un sistema de visión estereoscópica.

Un ejemplo de esto, son los sistemas láseres capaces de realizar el cálculo de distancias, pero estos sistemas pueden dar resultados erróneos en función del material sobre el que incida el haz. Otros sistemas existentes para el cálculo de distancias son los basados en radiación infrarroja (IR), un ejemplo comercial es Kinect o el modelo WaviXtion que está desarrollando Asus con la colaboración de Primesense, pero estos sistemas pueden no tener un buen funcionamiento al emplearse en exteriores durante días soleados, al ser parte del espectro electromagnético de la luz solar radiación infrarroja [4] [19] .

Así, el sistema de adquisición de imágenes escogido consta de dos cámaras alineadas milimétricamente y separadas una distancia fija, mediante las cuales se obtiene la información necesaria para posteriormente realizar el cálculo de contornos y profundidad.

4.2. Cámaras

El sistema de adquisición debe estar compuesto por un sistema de dos cámaras y estas deben cumplir una serie de restricciones:

- La separación entre cámaras debe ser similar a la existente entre los ojos de los seres humanos (entre 62 y 65mm) para obtener así resultados más acordes con la realidad.
- Ambas cámaras deben poseer las mismas características o muy similares.
- La alineación entre cámaras, como se mencionó anteriormente, debe ser milimétrica.

Las dos últimas restricciones son impuestas como motivo del posterior cálculo de la profundidad de las imágenes, ya que se realiza mediante algoritmos de correlación (ver capítulo 6). Estos algoritmos calculan la profundidad de cada pixel de la imagen (izquierda o derecha) en función de la variación que exista con el pixel correspondiente en la otra imagen.

Las cámaras que se han utilizado se pueden distinguir entre las usadas en la etapa inicial de pruebas y las escogidas para dar una solución definitiva.

4.2.1. Etapa de pruebas

Durante la etapa de pruebas se han empleado dos modelos de cámaras de diferentes características, para valorar que tipo de cámara era más adecuada para el desarrollo del proyecto.

En esta etapa las cámaras que se emplearon para el desarrollo del proyecto fueron las cámaras web Creative Live! Cam [20] y la cámara Minoru 3D [21]. A continuación se describen estas cámaras y se comentan los resultados obtenidos.

4.2.1.1. Cámara Creative

La primera implementación del sistema se realizó con dos cámaras web de la marca Creative, modelo Live! Cam VF0470.



Figura 4.1. Cámara web Creative Live! Cam VF0470.

Especificaciones técnicas:

- Sensor de imagen VGA CMOS con ajuste automático.
- Resolución de foto hasta 1,3 megapíxeles con interpolación.
- Resolución de video hasta 800x600 pixel con interpolación.
- Campo de visión de hasta 42 grados.
- Conexión mediante USB.

Los resultados obtenidos con este modelo de cámara no fueron muy alentadores por varios motivos.

Para cumplir con el requisito de alineación milimétrica entre cámaras, se diseñó un sistema que permitiese el ajuste en altura independiente de cada cámara (ver figura 4.2). Sin embargo, este sistema no ofrece una solución lo suficientemente óptima, por lo que se hizo necesario una post-calibración vía software, para intentar minimizar o incluso eliminar las diferencias de altura entre ambas cámaras. El proceso de calibración software se describe en el apartado 6.2.2.

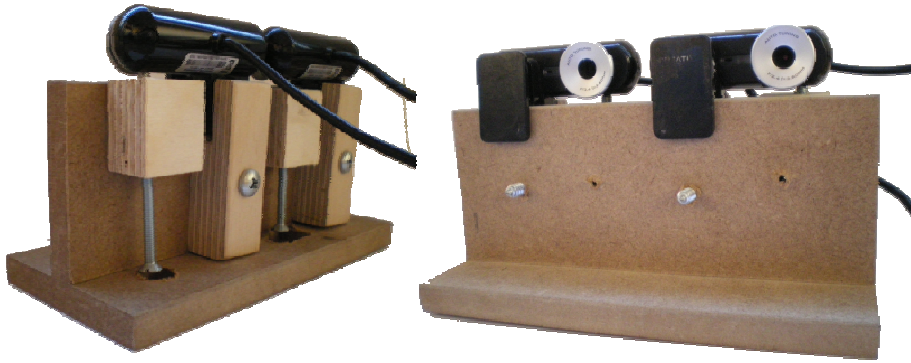


Figura 4.2. Sistema empleado para calibrar las cámaras Creative.

Tras la realización de las calibraciones vía hardware y vía software del sistema, se comprobó que la alineación milimétrica entre cámaras era un requisito alcanzable, sin embargo, la implementación de dos cámaras independientes no aporta una solución lo suficientemente robusta, ya que se hace necesaria la comprobación de la correcta alineación de las cámaras antes de cada uso y existe la posibilidad de producirse desalineaciones durante su uso.

A pesar de esto, el sistema compuesto por las dos cámaras Creative, tuvo que descartarse debido a la baja calidad del sensor. Esta comprobación se realizó tomando una imagen de la misma escena con cada cámara y comparando los valores de sus píxeles correspondientes. Obteniéndose como se puede ver en la figura siguiente que los valores obtenidos son muy dispares, llegando a obtenerse para el caso más desfavorable diferencias de hasta 60 en el valor de un píxel con su correspondiente en la otra imagen, lo cual es mucho teniendo en cuenta que los valores de los píxeles oscilan entre 0 y 255. Además, estas cámaras disponen de un control de la ganancia independiente, lo cual hace inviable la modificación de estos parámetros vía software.

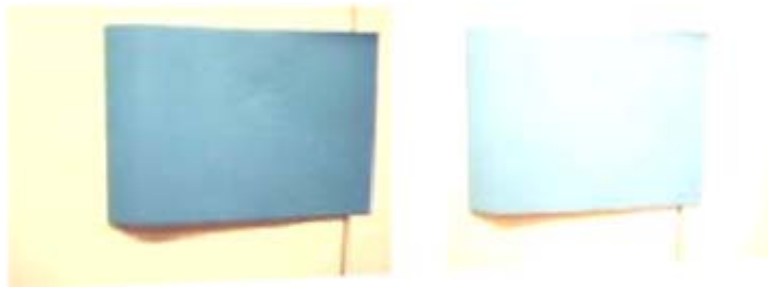


Figura 4.3. Imagen tomada por cámara Creative.

4.2.1.2. Cámara Minoru 3D

Para evitar el problema de calibración, se decidió hacer uso de un sistema que tuviese integradas ambas cámaras en una misma estructura. Para solventar el problema de los diferentes valores obtenidos con las dos cámaras, se buscó un sistema cuyas prestaciones en cuanto a la calidad de la imagen fuesen mayores.

De esta manera se encontró la cámara Minoru 3D, la cual tiene integrada ambas cámaras en un mismo sistema.

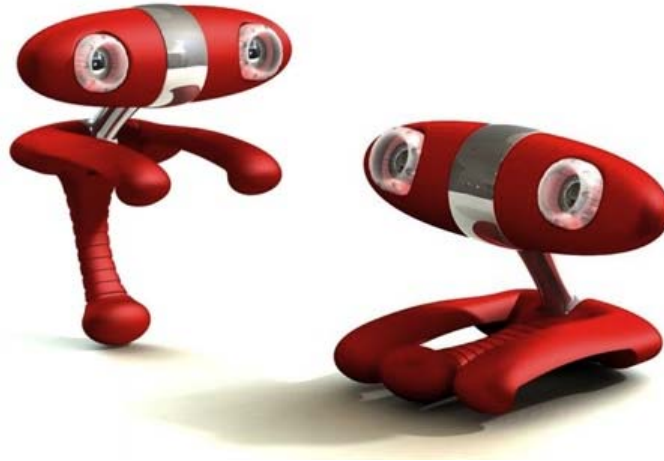


Figura 4.4. Cámara Minoru 3D.

Las especificaciones técnicas de la cámara Minoru 3D son:

- Conexión mediante USB 2.0 de alta velocidad.
- 2 x VGA 640x480 CMOS sensor.
- Separación entre cámaras: 62mm.
- 2 x lentes de ángulo ancho de alta calidad.
- Imagen de 800x600, 640x480, 352x288 y 320x240 pixeles.
- Velocidad de hasta 30 fps.
- Consumo de energía menor a 1,5 W en modo de funcionamiento.
- Pedestal Multi-posición.
- Imagen – Anáglifo 3D optimizado (rojo/cian), Imagen en Imagen (PiP) y 2D estándar.
- Micrófono USB incorporado.
- Compatible con Microsoft Windows XP y Microsoft Windows Vista.

Tras la realización de pruebas de calibración (ver apartado 6.2.2), para la cámara Minoru 3D, se concluyó que los ejes de ambos sensores se encontraban alineados milimétricamente, por lo que no es necesario realizar el proceso de calibración.

Para realizar la comprobación de las mismas características de las cámaras se realizó el mismo proceso que con las cámaras Creative, obteniéndose que con ambas cámaras se consiguen resultados muy similares.

4.2.1.3. Otras opciones

Además de las cámaras utilizadas existen gran variedad de cámaras en el mercado que podían haber sido empleadas para el desarrollo de este trabajo, de entre las cuales se destacan las siguientes cámaras por tener integradas en el mismo sistema ambas lentes.

- Zivora 3D Webcam: cámara web compuesta de dos sensores CMOS, 2xVGA con resolución de 640x480 y velocidad máxima de 60fps [22]
- Thanko Webcam 3D1: esta cámara web tiene una resolución de video de 320x240 a velocidad 15fps.
- Fujifilm FinePix Real 3D W3 Technology: cámara digital con dos sensores CCD de 10 megapíxeles [23]
- Inlife Easy 3D Stereo Digital Camera: cámara digital con sensores de 5 megapíxeles [24]



Figura 4.5. Cámaras Zivora 3D (izquierda) y Thanko Webcam 3D (derecha)



Figura 4.6. Fujifilm Finepix Real 3D W3 (izquierda) y Inlife Easy 3D (derecha).

4.2.2. Equipo utilizado

El equipo HMD utilizado para el sistema de salida consta también de dos cámaras que cumplen con las especificaciones marcadas por el sistema de adquisición de imágenes. De esta manera, el equipo final empleado para la adquisición de imágenes es también el HMD modelo ARVision-3D de la marca Trivisio, cuyas hojas de características se pueden ver en el anexo 2 [18].



Figura 4.7. HMD Trivisio ARVision 3D.

Algunas de las características más importantes de las cámaras que componen el HMD Trivio ARVSION3D son:

- Cámaras: 2x 752x480 (max 61fps), USB 2.0 (opcional IEEE1394).
- Sensor de las cámaras: 1/3" CMOS, barrido progresivo, obturador global.
- Potencia consumida: <1W cada cámara (5V, tomados de un puerto USB).
- Lentes: 6mm (0.9x vision), cambiables por el usuario, montura M12.
- Campo de visión: 40.6° diagonal (4:3, 32.7° (horiz), 24.5° (vert)).

CAPÍTULO 5. SISTEMA DE PROCESAMIENTO

5.1. Introducción

El sistema de procesamiento es el encargado de recibir las imágenes capturadas por el sistema de adquisición, procesarlas a través de algoritmos desarrollados (ver capítulo 6) y enviarlas al sistema de salida para mostrar al usuario el resultado obtenido.

Los dos principales factores a tener en cuenta a la hora de buscar un sistema de procesamiento adecuado para la aplicación que nos ocupa, son la portabilidad, autonomía y velocidad de procesamiento.

La portabilidad es una característica fundamental a la hora de permitir la movilidad del usuario. Hay dispositivos que cumplen con esta propiedad, tales como PDAs, teléfonos móviles o tabletPcs, pero suelen estar asociados a una velocidad de procesamiento insuficiente para realizar esta aplicación dentro de unos rangos de tiempos que puedan ser considerados aceptables como funcionamiento en tiempo real.

La velocidad de funcionamiento es un factor clave para el correcto funcionamiento de la aplicación desarrollada. Siendo el objetivo de esta aplicación, la búsqueda de una solución que permita al usuario recibir la información procesada con una sensación de continuidad suficiente para ser considerada tiempo real.

En la actualidad existen dispositivos con capacidad suficiente para albergar la aplicación desarrollada y dispositivos portables que cumplen óptimamente con la restricción de portabilidad. El problema es encontrar uno cuyo compromiso entre velocidad y portabilidad sea adecuado. Teniendo en cuenta estas restricciones parece obvio que el dispositivo que mejor encaja con estos requisitos es un ordenador portátil.

5.2. Equipos susceptibles de ser utilizados como sistema de procesamiento

La mayor restricción del sistema de procesamiento viene impuesta principalmente por la velocidad de procesamiento, como se expuso anteriormente. Por ello se realizó un estudio sobre las diferencias existentes en el tiempo de cómputo necesario para realizar los algoritmos desarrollados con ordenadores de diferentes características. En concreto se realizaron pruebas con un netbook SAMSUNG NC10 con procesador Intel ATOM N270 y 1GB de memoria RAM y un ordenador de sobremesa genérico con procesador Intel Core i7 y 4GB de memoria RAM.

En la siguiente figura se ilustran las diferencias existentes entre usar un dispositivo u otro.

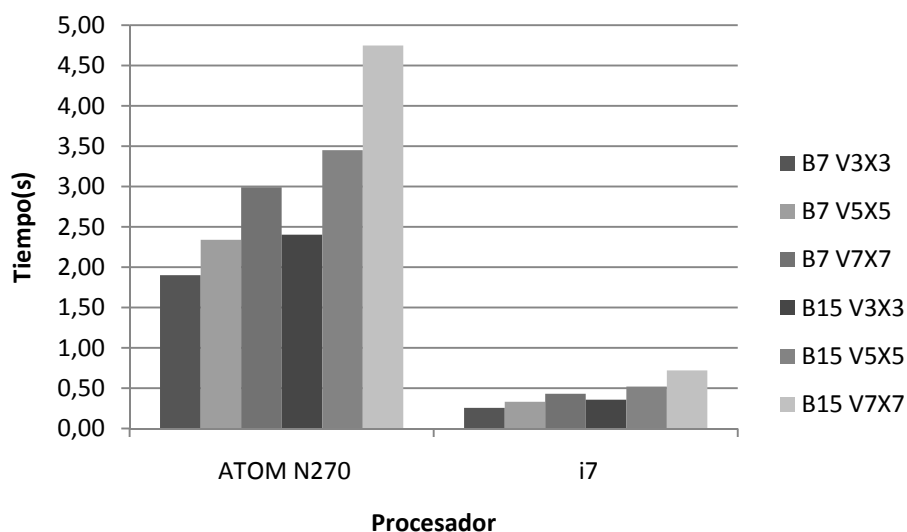


Figura 5.1. Gráfica comparativa tiempos de cómputo entre Intel i7 y ATOM N270.

Las diferencias entre el empleo de un procesador u otro son notables, llegando a procesarse hasta cinco imágenes por segundo con el Intel i7 y tan solo una por cada dos segundos con el ATOM para un tamaño de imagen de 120x160 píxeles. Una vez seleccionado el tipo de procesador es conveniente prestar atención a otros factores fundamentales para la elección del sistema de procesamiento, como son el peso, autonomía y tipos y número de entradas con el exterior.

El peso del ordenador es conveniente que sea lo más reducido posible para hacer el conjunto del sistema lo más portable posible.

La alimentación de todos los dispositivos electrónicos que componen el sistema, se realiza a través del sistema de procesamiento. Por ello, es necesario que la capacidad de las baterías sea lo mayor posible para dotar así al sistema final de mayor autonomía.

Como se mencionó en apartados anteriores el sistema de adquisición de datos y el de salida están integrados ambos en un casco de realidad virtual, el cual es necesario conectar con el sistema de procesamiento. Para ello, el sistema de procesamiento debe constar de las conexiones necesarias para la conexión con el casco de realidad virtual (HMD).

5.3. Equipo utilizado

Como ya se definió, el sistema de procesamiento es un nexo de unión entre los sistemas de salida y de adquisición de imágenes y se requiere por tanto que tenga 3xUSB (2x cámara y 1xalimentación del HMD) y una entrada DVI para el envío de datos al HMD.



Figura 5.2. Conectores del HMD ARVision 3D.

Las entradas para video de alta definición DVI no son muy comunes en equipos portátiles, siendo más comunes las entradas HDMI (*High Definition Multimedia Interface* o interfaz multimedia de alta definición en español). Teniendo en cuenta esto, el equipo seleccionado requerirá de un adaptador de HDMI a DVI para la conexión con las gafas de realidad virtual.



Figura 5.3. Adaptador HDMI a DVI.

La elección del tipo de procesador está basada en el estudio hecho en los apartados anteriores. Siendo el equipo seleccionado un ordenador portátil de la marca HP modelo 14,5" Envy 14-1180es Beats Edition Intel Core i7 720QM. A continuación se muestran alguna de sus características más relevantes.

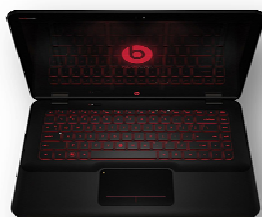


Figura 5.4. Ordenador HP Envy 14-1180es.

- Tipo de procesador: Intel i7-720QM.
- Velocidad de procesador 1,6 GHz.
- Memoria RAM: 4GB.
- Número de celdas: 8 celdas.
- Tarjeta gráfica: ATI Mobility Radeon HD 5650.
- Peso: 2,5kg.

5.4. Montaje final del sistema

El montaje final del sistema, supone la integración de todos los sistemas descritos en los capítulos anteriores. Esta integración se realiza a través del ordenador portátil que recibe las imágenes captadas por las cámaras del HMD Trivisio ARVision3D y las procesa para finalmente enviar los resultados a las micropantallas que posee este casco de realidad virtual.

El sistema de procesamiento será transportado o bien en una mochila o bien en un cinturón y el sistema de adquisición de imágenes y el sistema de salida están ambos integrados en el HMD.

CAPÍTULO 6. ALGORITMOS

6.1. Lenguaje de programación

Para la realización de este proyecto se ha empleado la herramienta informática MatLab. Esta elección se ha debido, en parte, a que algunos trabajos y estudios previos se realizaron bajo este entorno.

MatLab (abreviatura de “Matrix Laboratory”), es un software matemático que ofrece un entorno de desarrollo integrado con un lenguaje propio (lenguaje M). MatLab es un lenguaje de alto nivel para cálculos técnicos y numéricos, que permite resolver problemas de técnicos de manera más rápida que con los lenguajes de programación tradicionales.

MatLab dispone de un código básico y de varias cajas de herramientas especializadas, denominadas *toolboxes*, que amplían sus capacidades. Entre estas cabe destacar *Image Processing Toolbox* la cual ha sido de gran utilidad en el desarrollo de los algoritmos ya que proporciona un amplio conjunto de algoritmos estándar de referencia y herramientas gráficas para el procesamiento de imagen, análisis, visualización y desarrollo de algoritmos. Muchas de estas aplicaciones son multiproceso para aprovechar las ventajas de los procesadores multinúcleo o multiprocesador.

Además es interesante mencionar que MatLab dispone de una herramienta llamada GUIDE (*Graphical User Interface Development Environment*), la cual permite la creación de interfaces gráficas de usuario [25].

6.2. Diseño de los algoritmos

A partir de este punto se procederá a describir los algoritmos que se han desarrollado para intentar solventar o reducir los problemas derivados de carecer de sensación de profundidad. Por lo tanto, los algoritmos empleados que son descritos en este capítulo están destinados a aportar información extra sobre la profundidad de los objetos del entorno.

Para poder entender mejor el conjunto de algoritmos desarrollados, es conveniente prestar atención al diagrama de bloques general del algoritmo, el cual se muestra a continuación en la figura 6.1.

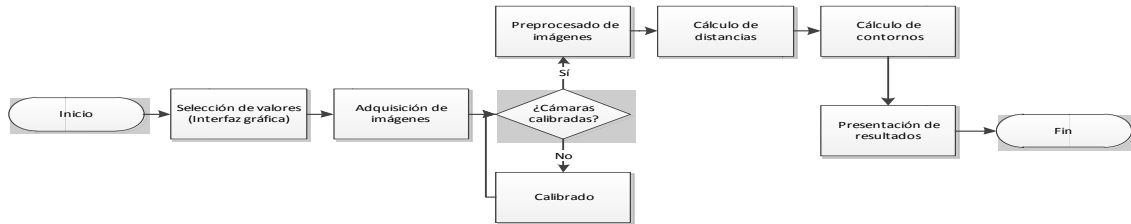


Figura 6.1. Diagrama general de los algoritmos desarrollados.

6.2.1. Interfaz gráfica

La aplicación desarrollada está destinada para ser probada por optometristas con pacientes reales. Por consiguiente, es necesario el desarrollo de un entorno lo más amigable y versátil posible para facilitar la realización de estas pruebas.

Esta interfaz gráfica ha sido desarrollada haciendo uso de la herramienta GUIDE de MatLab y permite, sin tener conocimientos de MatLab, la modificación de diferentes parámetros.

- Valores necesarios para el cálculo de la correlación (barrido y ventana).
- Tamaño de la imagen sobre la que se realizan los algoritmos.
- Color y grosor de los contornos.
- Rangos a los que se considera una distancia u otra.
- La posibilidad de mostrar los resultados a color, a escala de grises o solo los contornos.
- Realizar el cálculo sobre un par estéreo tomado previamente o haciendo uso de las webcam.

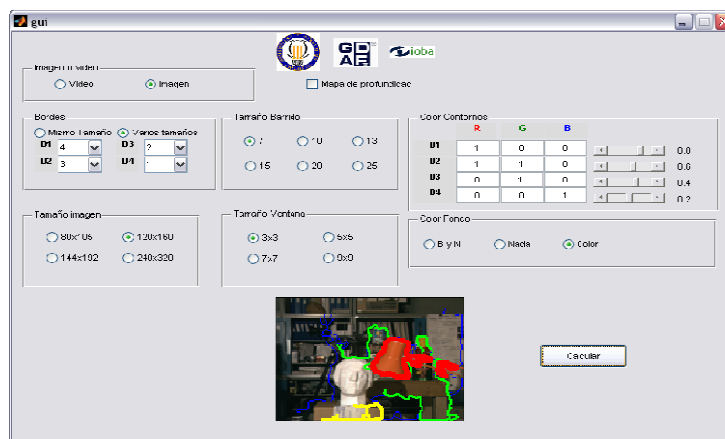


Figura 6.2. Interfaz gráfica desarrollada.

6.2.2. Calibración mediante FFT

Como se mencionó en el apartado 1.5. *Visión Binocular*, en un sistema estereoscópico debe existir coordinación estrictamente milimétrica entre ambas cámaras. De aquí la necesidad de hacer una calibración previa del sistema.

El método de calibración de las cámaras está basado en la transformada rápida de Fourier (FFT). La transformada rápida de Fourier es un eficiente algoritmo que permite calcular la transformada de Fourier discreta y su inversa. Siendo ésta de mucha importancia en una gran variedad de aplicaciones.

Mediante la transformada rápida de Fourier se puede realizar el proceso de búsqueda de un patrón conocido en otra imagen de manera rápida y efectiva. Aunque existe el problema de que puede encontrar el patrón que se busca en cualquier figura que lo contenga.

Para comprender esto mejor conviene ver el siguiente ejemplo, donde se busca en un texto (parte superior izquierda de la imagen) un patrón consistente en una letra "A" (parte superior derecha de la imagen).

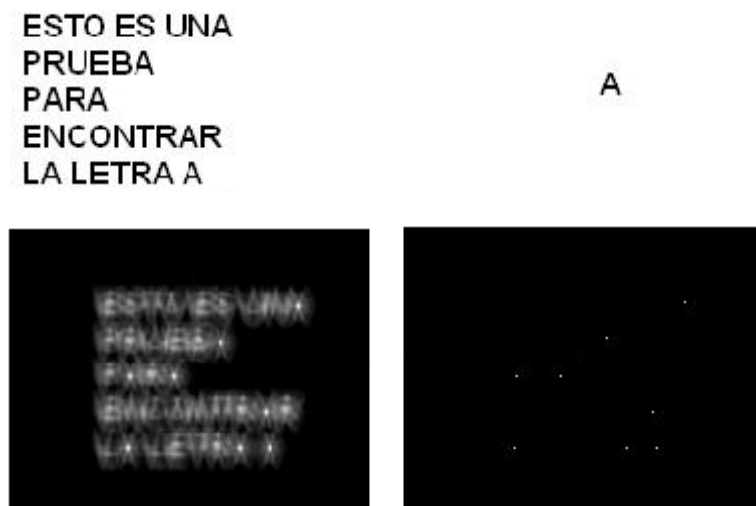


Figura 6.3. Ejemplo de búsqueda de un patrón usando FFT.

Tras aplicar el algoritmo de la búsqueda de un patrón mediante la FFT, el resultado que se obtiene es el mostrado en la parte inferior de la figura 6.3., donde la parte de la izquierda muestra el espectro de la búsqueda del patrón de la figura anterior y en la parte derecha se muestra umbralizado el espectro para resaltar tan solo los puntos de máxima correlación, los cuales coinciden con las aes buscadas.

Sin embargo, cuando la búsqueda de la letra "A" se realiza en una imagen que además de contener texto contiene un recuadro negro, el cual puede contener el patrón buscado, los resultados ya no son tan buenos, debido a que la FFT encuentra como punto de máxima correlación el recuadro negro (figura 6.4).

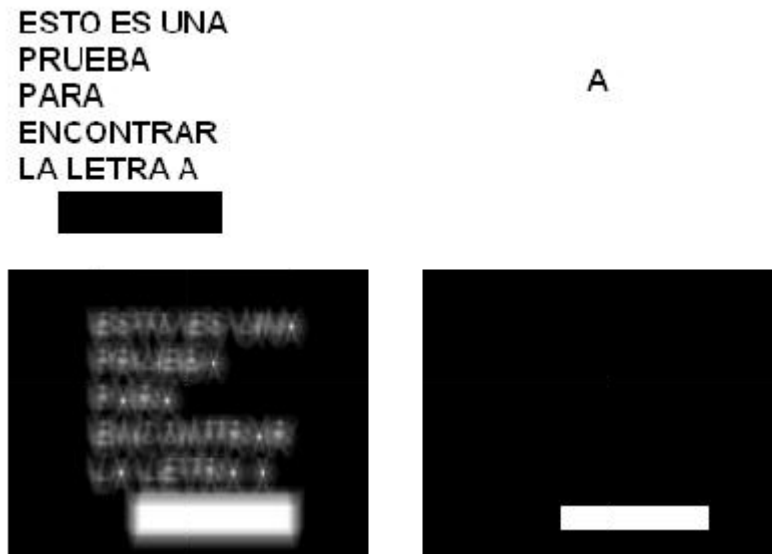


Figura 6.4. Ejemplo búsqueda de un patrón usando FFT.

Teniendo en cuenta tanto los convenientes e inconvenientes, el método que se ha escogido para la calibración de las cámaras está basado en la búsqueda de un mismo patrón en las imágenes captadas por ambas cámaras mediante la transformada rápida de Fourier. El patrón buscado es un cuadro negro sobre un fondo blanco, por lo que en el proceso de calibración es necesario enfocar con las cámaras una pared blanca con un cuadro negro.

El proceso de funcionamiento del algoritmo de calibración sigue el siguiente diagrama de flujo. Para ver el algoritmo de calibración, ver el anexo 1 de esta memoria.

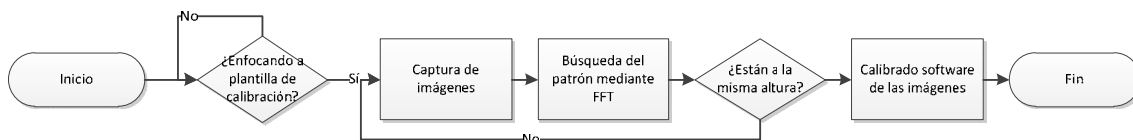


Figura 6.5. Diagrama de flujo de la calibración.

En el diagrama de flujo anterior, se deduce que los pasos del proceso de calibración son:

- 1- Búsqueda del patrón (cuadro negro) en ambas imágenes del par estereoscópico mediante la transformada rápida de Fourier (FFT).
- 2- Obtención de la fila en la que se ha localizada el patrón en cada imagen.
- 3- Ajuste mediante software de las imágenes para minimizar o eliminar la diferencia de altura entre éstas.
- 4- Se repiten los pasos anteriores hasta obtener que el patrón buscado se encuentra en la misma fila para ambas imágenes.

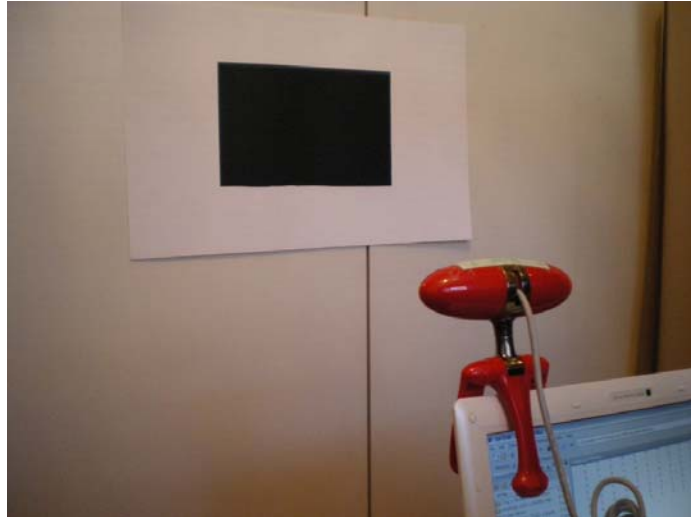


Figura 6.6. Proceso de calibración con cámara Minoru 3D.

Tras las pruebas realizadas con la cámara Minoru 3D, se comprobó que no era necesario el calibrado vía software de las cámaras al estar perfectamente alineadas de fábrica.

6.2.3. Adquisición de imágenes

El primer paso una vez realizada la calibración del sistema, es la captura del par estereoscópico para realizar posteriormente el tratamiento de estas imágenes. Las imágenes del par estéreo deben de presentar un ligero desplazamiento debido a la separación de las cámaras, la cual conviene que sea similar a la existente en los ojos de los seres humanos.

La adquisición del par estereoscópico se realiza con el uso de las cámaras mencionadas en el capítulo cuarto. Además, como se mencionó en el capítulo introductorio, para la visión binocular es necesario tener dos imágenes pertenecientes a la misma escena, por lo que la adquisición del par estéreo debe de ser simultánea, para así evitar cambios debidos a los movimientos de los objetos o del propio sistema de captura.

La herramienta de procesamiento de imágenes de MatLab consta de una serie de funciones que permiten realizar la captura de fotogramas desde un objeto de video.

En el anexo 1 se puede examinar los algoritmos completos usados en la captura de las imágenes.

6.2.4. Preprocesamiento

Un factor de complicación para el procesado estéreo de imágenes es que la intensidad de los correspondientes píxeles de la imagen estéreo puede ser diferente. Esta variación puede ser causada por diferentes factores:

- Características diferentes de los sensores de la cámara izquierda y derecha, como puede ser el brillo o el contraste.
- Debido a condiciones de iluminación de la posición de la cámara.

Si se hace uso de imágenes en bruto, es necesario usar una medida de similitud invariante a la iluminación. Debido a que las mediciones invariantes son computacionalmente más caras, el preprocesado de las imágenes de entrada es a menudo aplicado para reducir las diferencias de iluminación de antemano.

La propuesta más popular es la convolución de las imágenes de entrada con el núcleo de la Laplaciana de una Gaussiana [28][29]. En las siguientes figuras se pueden ver las diferencias existentes de aplicar diferentes filtros. En la figura 6.7. se representan los mapas de disparidad (ver apartado 6.2.5.) y en la figura 6.8. el resultado final una vez calculados los contornos, para la imagen de la referencia [26].

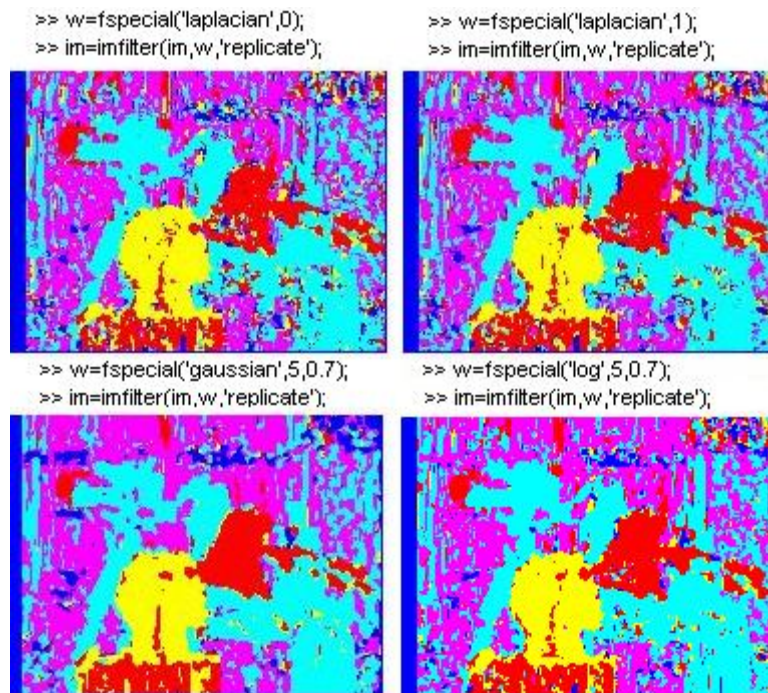


Figura 6.7. Mapa de disparidad tras aplicación de diferentes filtros.

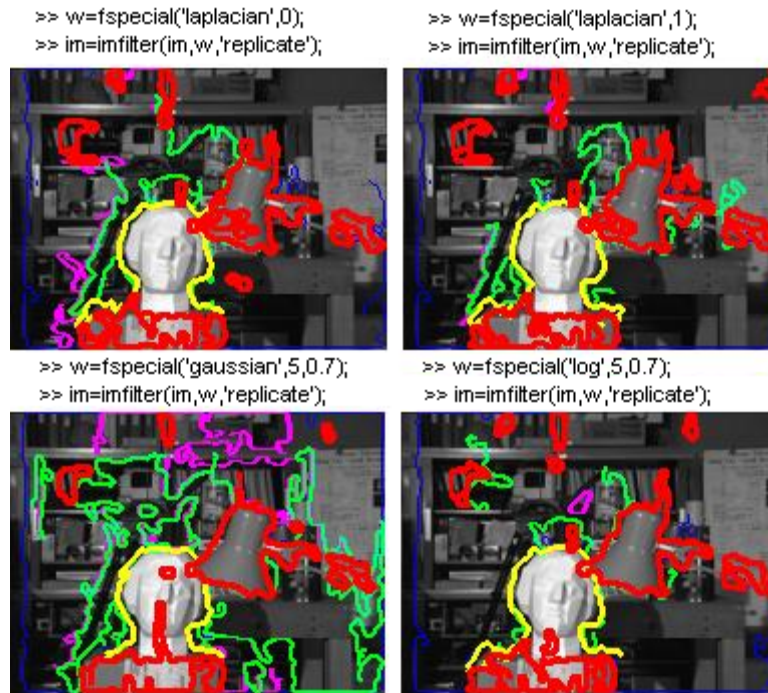


Figura 6.8. Resultado final tras aplicación de diferentes filtros.

Los tiempos de cómputo empleados en el preprocesamiento de las imágenes oscilan entre 1 y 3 milisegundos para cada imagen de tamaño 120x160, lo cual puede considerarse despreciable frente a los tiempos empleados en el resto de algoritmos.

6.2.5. Cálculo de la distancia

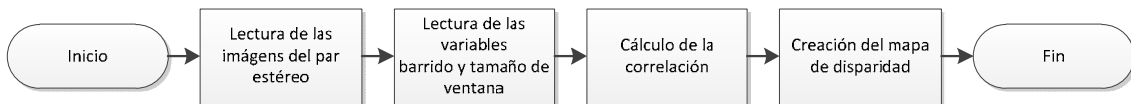


Figura 6.9. Diagrama de flujo del algoritmo para el cálculo de la distancia.

Una vez capturado y preprocesado el par de imágenes estéreo, el siguiente paso es el cálculo de las distancias de los objetos presentes en las imágenes. El cálculo de las distancias está basado en el cálculo de la disparidad de los píxeles de la imagen.

Los algoritmos estéreo de cálculo de la disparidad tienen por objetivo obtener la profundidad de todos los píxeles de las imágenes del par estéreo. Hallando las parejas de puntos correspondientes que son proyección del mismo punto del espacio, para todos los puntos en cada una de las imágenes. De esta forma se obtiene un mapa denso de disparidades.

Los algoritmos de cálculo de disparidad asumen, por simplicidad, que los planos de las imágenes en cada cámara son paralelos entre sí, y paralelos a la dirección en la cual existe el desplazamiento entre las imágenes. De esta forma el correspondiente de un pixel en la fila y_L de la imagen izquierda se encuentra en la fila $y_R = y_L$ de la imagen derecha.

Para hallar el pixel correspondiente en la otra imagen, se imponen una serie de restricciones o hipótesis.

- Restricción epipolar: el correspondiente de un punto debe estar en la línea epipolar del punto en la otra imagen.
- Restricción de orden: los homólogos de un objeto aparecen en el mismo orden sobre las epipolares correspondientes.
- Restricción de unicidad: un punto de la imagen tiene como máximo una única correspondencia en la otra imagen del par.
- Restricción de semejanza: las características de los puntos en las imágenes (intensidad, color, saturación) no deben de cambiar mucho.

El fin del cálculo de la disparidad es la creación de lo que se conoce como mapa de profundidad, siendo éste una nueva imagen del mismo tamaño que las imágenes del par estéreo cuyo valor de los pixeles varía en función de la distancia a la que se encuentran (disparidad), asignando diferentes niveles de grises o colores según un criterio preestablecido. En la figura 6.10. y figura 6.11. se observan los mapas de profundidad de una misma escena, a color y en niveles de gris respectivamente.



Figura 6.10. Mapa de profundidad a color.

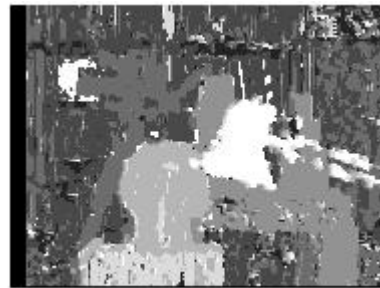


Figura 6.11. Mapa de profundidad en escala de grises.

Los mapas de disparidad anteriores se han obtenido del par de imágenes estéreo “*head and lamp*” de la Universidad de Tsukuba de Japón [26], las cuales se muestran a continuación en la Figura 6.12.



Figura 6.12. Par de imágenes estéreo Tsukuba[26].

En el siguiente apartado se describen diferentes métodos matemáticos para el cálculo de la disparidad basados en la correlación.

6.2.5.1. Métodos de correlación

La correlación es una variable estadística adimensional con la que se mide la dependencia lineal de dos variables. Este es el método matemático que se ha utilizado para establecer la correspondencia entre los píxeles homólogos. El cálculo de la correlación constituye una de las etapas más importantes de un algoritmo de visión estéreo, además de ser la que mayor esfuerzo de cálculo requiere.

La comparación mediante técnicas de correlación utiliza la correlación cruzada para determinar la similitud de imágenes [27]. El cálculo de la correlación cruzada de la imagen izquierda del par estéreo (usada como patrón o *template*) y la imagen derecha del par estéreo, puede llevarse a cabo aplicando uno de los diferentes métodos matemáticos existentes para la consecución de esta tarea. Entre estos métodos se destacan:

- Suma de diferencias absolutas (SDA).

$$SAD = \sum_{i,j} |I_T(x+i, y+j) - I_D(x+i-d, y+j)|$$

- Suma de diferencias cuadradas (SSD).

$$SSD = \sum_{i,j} (I_T(x+i, y+j) - I_D(x+i-d, y+j))^2$$

- Suma normalizada de diferencias cuadradas (NSSD).

$$NSSD = \frac{\sum_{i,j} (I_T(x+i, y+j) - I_D(x+i-d, y+j))^2}{\sqrt{\sum_{i,j} (I_T(x+i, y+j))^2 \sum_{i,j} (I_D(x+i-d, y+j))^2}}$$

- Correlación normalizada (NCC).

$$NCC = \frac{\sum_{i,j} (I_T(x+i, y+j) \cdot I_D(x+i-d, y+j))}{\sqrt{\sum_{i,j} (I_T(x+i, y+j))^2 \sum_{i,j} (I_D(x+i-d, y+j))^2}}$$

Sin embargo estos métodos se ven muy influenciados por la diferencia de ganancia de las cámaras, por lo que sería necesario aplicar los preprocesamientos comentados en el apartado 6.2.4 para resolver esta problemática. En caso de no usar estos preprocesamientos es necesario emplear la resta de la media de la ventana, para ello se emplearían los siguientes métodos.

- Suma de diferencias absolutas con media cero (ZSAD).

$$ZSAD = \sum_{i,j} |(I_T(x+i, y+j) - \bar{I}_T) - (I_D(x+i-d, y+j) - \bar{I}_D)|$$

- Suma de diferencias cuadradas con media cero (ZSSD).

$$ZSSD = \sum_{i,j} ((I_T(x+i, y+j) - \bar{I}_T) - (I_D(x+i-d, y+j) - \bar{I}_D))^2$$

- Suma normalizada de diferencias cuadradas con media cero (ZNSSD).

$$ZNSSD = \frac{\sum_{i,j} ((I_T(x+i, y+j) - \bar{I}_T) - (I_D(x+i-d, y+j) - \bar{I}_D))^2}{\sqrt{\sum_{i,j} (I_T(x+i, y+j) - \bar{I}_T)^2 \sum_{i,j} (I_D(x+i-d, y+j) - \bar{I}_D)^2}}$$

- Correlación normalizada con media cero (ZNCC).

$$ZNCC = \frac{\sum_{i,j} ((I_T(x+i,y+j) - \bar{I}_T) \cdot (I_D(x+i-d,y+j) - \bar{I}_D))^2}{\sqrt{\sum_{i,j} (I_T(x+i,y+j) - \bar{I}_T)^2 \sum_{i,j} (I_D(x+i-d,y+j) - \bar{I}_D)^2}}$$

En el trabajo realizado por el GDAF en 2010 [2] se hace un estudio comparativo de los tiempos de cómputo de los métodos anteriores, de donde se concluye que los métodos que realizan la normalización del pixel requieren un esfuerzo computacional bastante mayor.

El método empleado para la obtención del mapa de disparidad ha sido el de la suma de diferencias absolutas (SDA), por ser el método que requiere realizar un menor número de operaciones y por tanto tiene menor coste computacional.

6.2.5.2. Cálculo de la correlación

El principio general de funcionamiento de la correlación consiste en comparar una determinada vecindad (ventana) de un pixel de la imagen patrón o de referencia con una vecindad de las mismas características de cada uno de los pixeles de la imagen a comparar. Como se explicó en el capítulo introductorio, las imágenes captadas por un sistema de visión binocular, tienen en común gran parte de la escena, pero no toda ya que los extremos laterales son exclusivos de cada imagen. Además, como se mencionó en apartados anteriores se asume que la disparidad es únicamente horizontal, por lo que parece lógico pensar que la búsqueda del homólogo de un pixel de la imagen de referencia se restringe sólo a un número limitado y predefinido de píxeles situados en la misma fila de la otra imagen. A este término lo llamaremos barrido.

Conviene advertir que esta operación no se realiza para todos los pixeles de la imagen, sino que existen unas limitaciones introducidas por las variables de la correlación (tamaño de ventana y barrido). En cuanto a la dimensión vertical de la matriz, tanto en la parte superior como en la inferior, la única restricción existente es el tamaño de la ventana de vecindad. Sin embargo, en la dimensión horizontal, además de existir esta limitación en ambos extremos, hay que añadir una limitación en la parte izquierda equivalente al tamaño del barrido.



Figura 6.13. Representación del área total en el que se calcula correlación.

La comprensión de la figura anterior en la que se muestra el área total en la que se realiza la comparación de los píxeles se hace más sencilla con la siguiente figura en la que se representa el proceso de búsqueda del homólogo de un pixel.

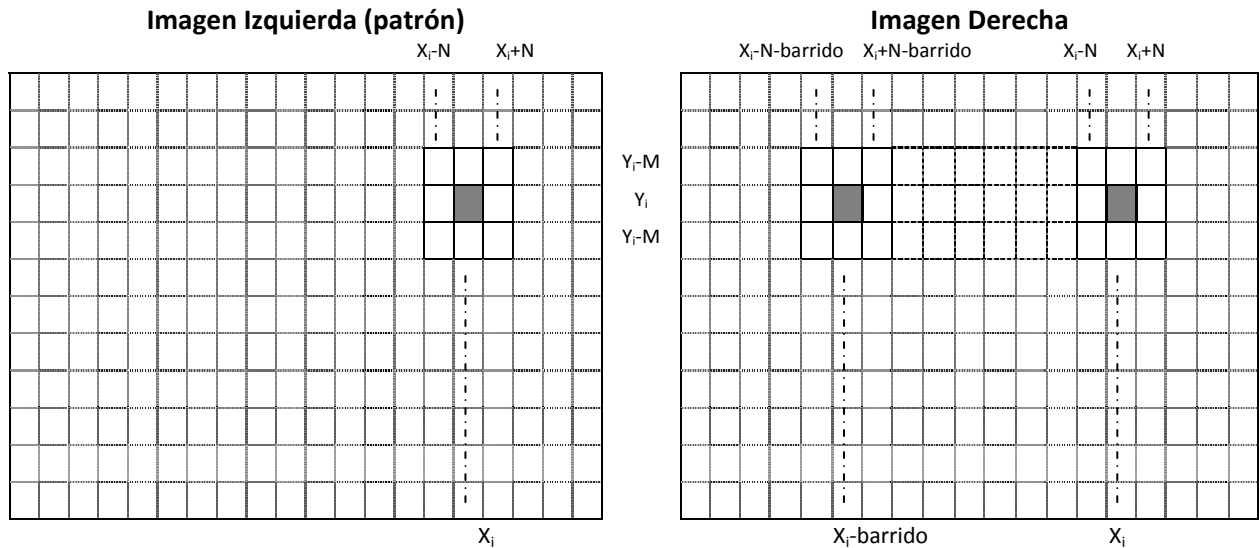


Figura 6.14. Fase de búsqueda de un pixel homólogo.

Se deduce que en la fase de búsqueda del homólogo de cada pixel, el número de correlaciones calculadas es igual al del tamaño de barrido seleccionado. Este conjunto de valores alcanza un valor extremo (mínimo o máximo según criterio) cuando la vecindad comparada es la más parecida. El desplazamiento al cual es encontrado el valor extremo de disparidad, es el que se guarda como válido para ese pixel y está relacionado directamente con la profundidad del pixel.

La efectividad de la búsqueda de los homólogos de un pixel se ve influenciada por la riqueza de los detalles de la imagen, como son texturas o presencia de discontinuidades, lo cual puede dar como resultado fallos de emparejamiento. La buena elección de los parámetros barrido y tamaño de la ventana de vecindad tiene un papel fundamental en la minimización de este efecto. Un tamaño de ventana demasiado pequeña puede capturar demasiada poca información de la imagen, mientras que por el contrario un tamaño de ventana demasiado grande puede producir una pérdida de definición de los objetos de la escena.

6.2.5.3. Tiempo de cómputo

El mayor problema de estos algoritmos es el alto coste computacional que tienen, haciendo difícil la ejecución del código en tiempo real.

Los tiempos de cómputo de la correlación varían en función de múltiples factores como son el tamaño de la imagen, el tamaño de ventana y barrido empleados para el cálculo de la correlación o el tipo ordenador empleado.

En las siguientes gráficas se muestra la variación de los tiempos de cómputo necesarios para el cálculo de la correlación para diferentes valores de barrido, tamaño de la ventana y tamaño de la imagen haciendo uso de un ordenador con procesador Intel i7.

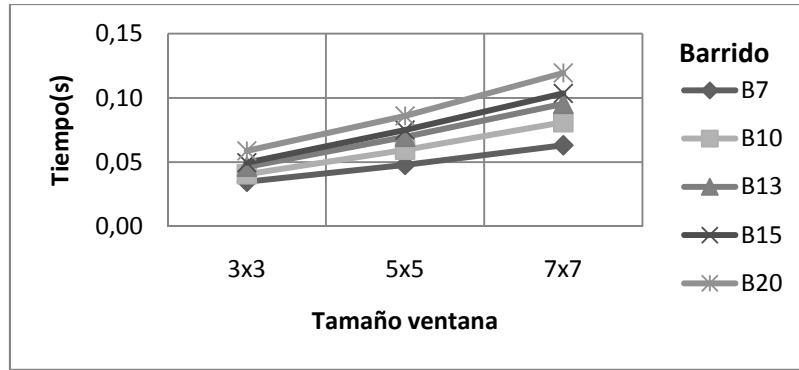


Figura 6.15. Tiempos de cómputo de la correlación en imagen 64x64.

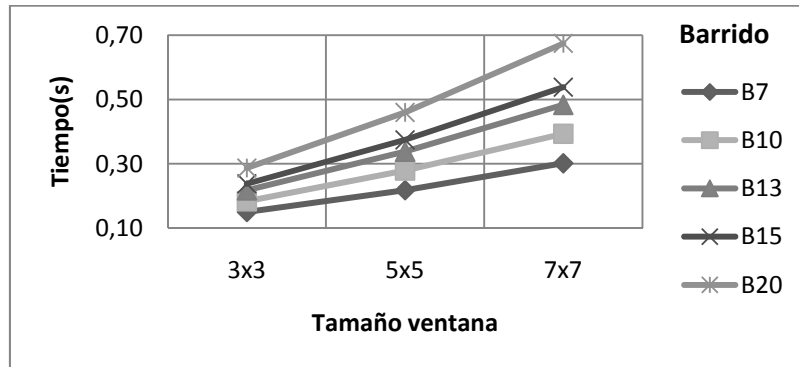


Figura 6.16. Tiempos de cómputo de la correlación en imagen 128x128.

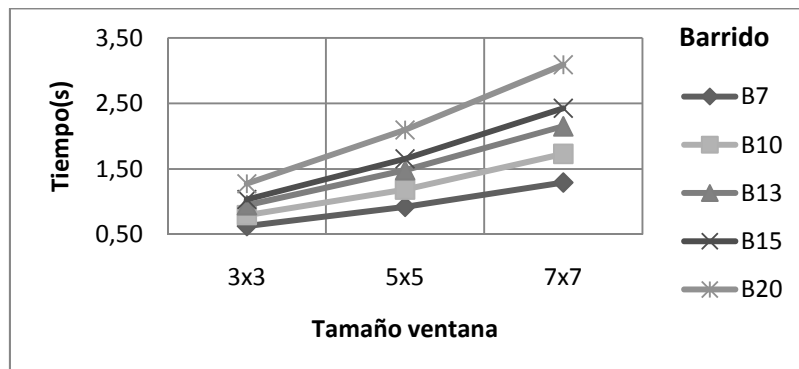


Figura 6.17. Tiempos de cómputo de la correlación en imagen 256x256.

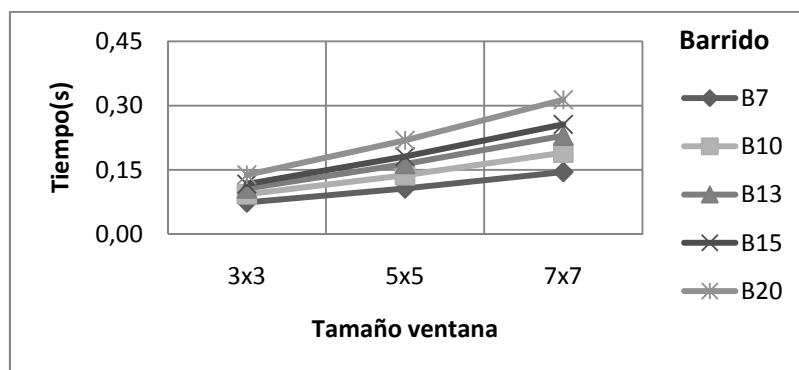


Figura 6.18. Tiempos de cómputo de la correlación en imagen 80x105.

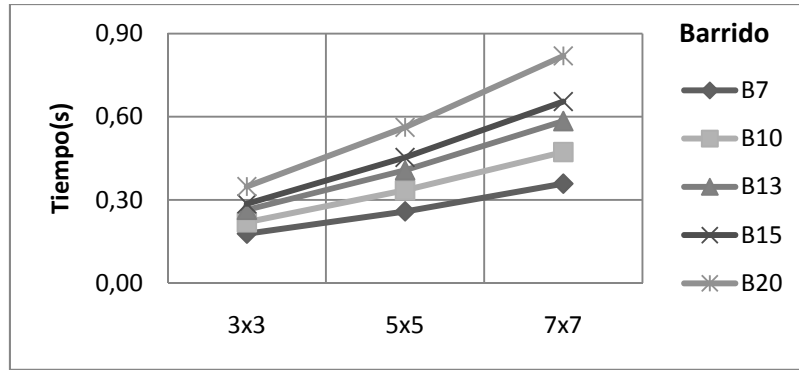


Figura 6.19. Tiempos de cómputo de la correlación en imagen 120x160.

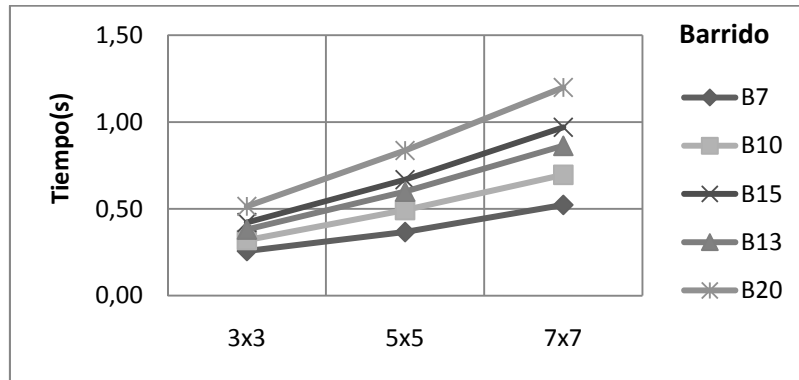


Figura 6.20. Tiempos de cómputo de la correlación en imagen 144x192.

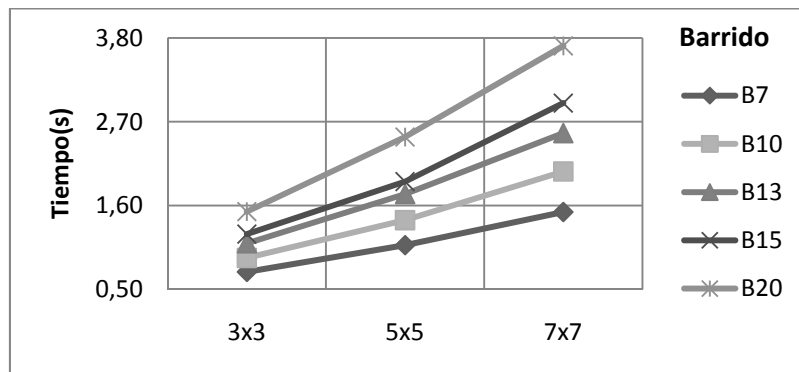


Figura 6.21. Tiempos de cómputo de la correlación en imagen 240x320.

6.2.6. Cálculo de contornos

La obtención de los contornos y la asignación de un color o grosor a estos en función de la distancia es el objetivo final de este proyecto. El coste computacional del algoritmo para el cálculo de contornos es variable en función del entorno y por supuesto del tamaño de la imagen adquirida, siendo el coste mayor en entornos en los que es necesario representar mayor cantidad de contornos.

La *toolbox Image Processing* de MatLab dispone de una función para el cálculo de contornos que es de gran importancia en el desarrollo de este apartado. Esta es la función *edge*, la cual soporta seis métodos diferentes de detección de contornos (Prewitt, Roberts, Sobel, Canny, Laplaciano-Gaussiano y *zero-cross*) y ofrece la posibilidad de fijar un umbral de sensibilidad en la detección de estos [25].

Los resultados que se obtienen al aplicar estos métodos, son bastante similares. Por lo que no existen diferencias suficientemente apreciables como para escoger un método en lugar de otro. Finalmente se optó por utilizar el método de Canny con un umbral que permita evitar el cálculo de contornos de objetos que no sean de relevancia.

Debido a la gran variedad de problemas de visión y los efectos de estos se ha optado por dos métodos diferentes para representar la información relativa a la distancia de los objetos:

- **Asignación de diferentes colores en función de la distancia.**

En este método los colores del contorno son asignados en función de la distancia a la que se encuentran los diferentes objetos.

En los ejemplos mostrados en este documento los colores de los contornos varían de más próximo a más lejano de la siguiente manera.

- Rojo.
- Amarillo.
- Turquesa.
- Magenta.
- Azul.

Esta asignación de colores no está basada en ningún estudio sobre que colores son más fáciles de distinguir por los pacientes, ya que en función del tipo y grado de enfermedad sufrida, puede ser más óptimo el uso de unos colores u otros. Un claro ejemplo es la confusión de colores debida al daltonismo.

En las siguientes figuras, se muestra un ejemplo de cómo varían los colores de los contornos en función de la distancia de los objetos.

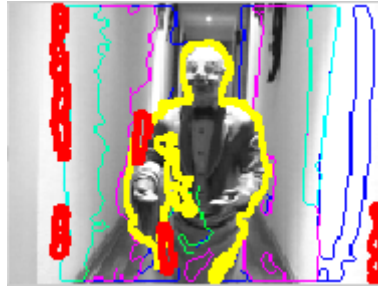


Figura 6.22. Representación de contornos a color, distancia 1.



Figura 6.23. Representación de contornos a color, distancia 2.

- **Variación del grosor del contorno en función de la distancia.**

Este método se diferencia con el anterior en que los contornos a parte de variar su color en función de la distancia varían también su grosor. Esta solución puede ser interesante tanto para pacientes que no distinguen bien los colores como para pacientes que aunque si distinguen colores, esta información extra les ayuda a procesar más rápida y eficientemente la información.

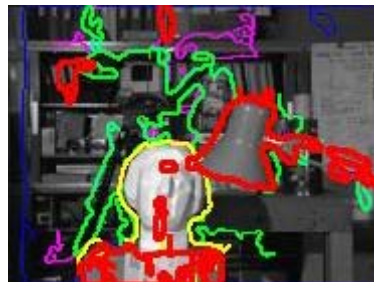


Figura 6.24. Representación contornos a color con diferentes grosores.

Los contornos son dilatados mediante la función *imdilate* de la *toolbox* de *Image Processing*. La máscara que se ha empleado para ampliar los contornos de las imágenes mostradas en este documento se muestran a continuación, aunque existe la posibilidad de variar el grosor de cada contorno desde la interfaz gráfica desarrollada.

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Figura 6.25. Máscara empleada en dilatación del contorno más próximo.

1	1	1
1	1	1
1	1	1

Figura 6.26. Máscara para dilatación del segundo contorno.

1	1
1	1

Figura 6.27. Máscara para dilatación del tercer contorno.

0	0	0
0	1	0
0	0	0

Figura 6.28. Máscara para dilatación cuarto y quinto contorno.

El cálculo de los contornos se ha implementado de dos maneras diferentes. En la primera de ellas, se calcula el contorno haciendo uso del mapa de profundidad de la imagen y se le asigna un color en función del valor de la disparidad. La otra opción es el cálculo de los contornos de la imagen original y la asignación del color correspondiente a ese pixel en el mapa de profundidad.

En el siguiente apartado, se detallan más estos métodos y se muestran los resultados obtenidos con cada uno. Para hacer más sencilla la comprensión de estos métodos, la explicación del método se realiza apoyándose en un ejemplo del cálculo del contorno para unos pares estéreos tomados con la cámara Minoru 3D.

6.2.6.1. Cálculo contornos usando mapa de disparidad

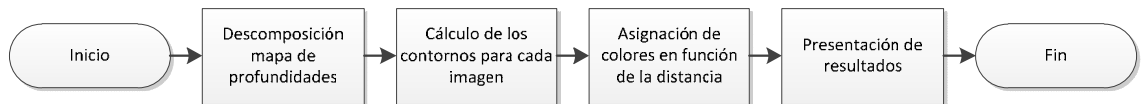


Figura 6.29. Diagrama de flujo del cálculo de contornos usando mapa de disparidad.

El problema del cálculo de los contornos radica fundamentalmente en diferenciar que valor se le asocia a cada pixel del contorno. Esto se ha logrado descomponiendo el mapa de disparidad de la imagen en diferentes imágenes para cada rango de distancia.



Figura 6.30. Mapa de profundidad de una imagen tomada con Minoru 3D.

Al umbralizar el mapa de disparidad de la figura 6.30 para cada rango de profundidad, se obtiene como resultado las imágenes de la figura 6.31.

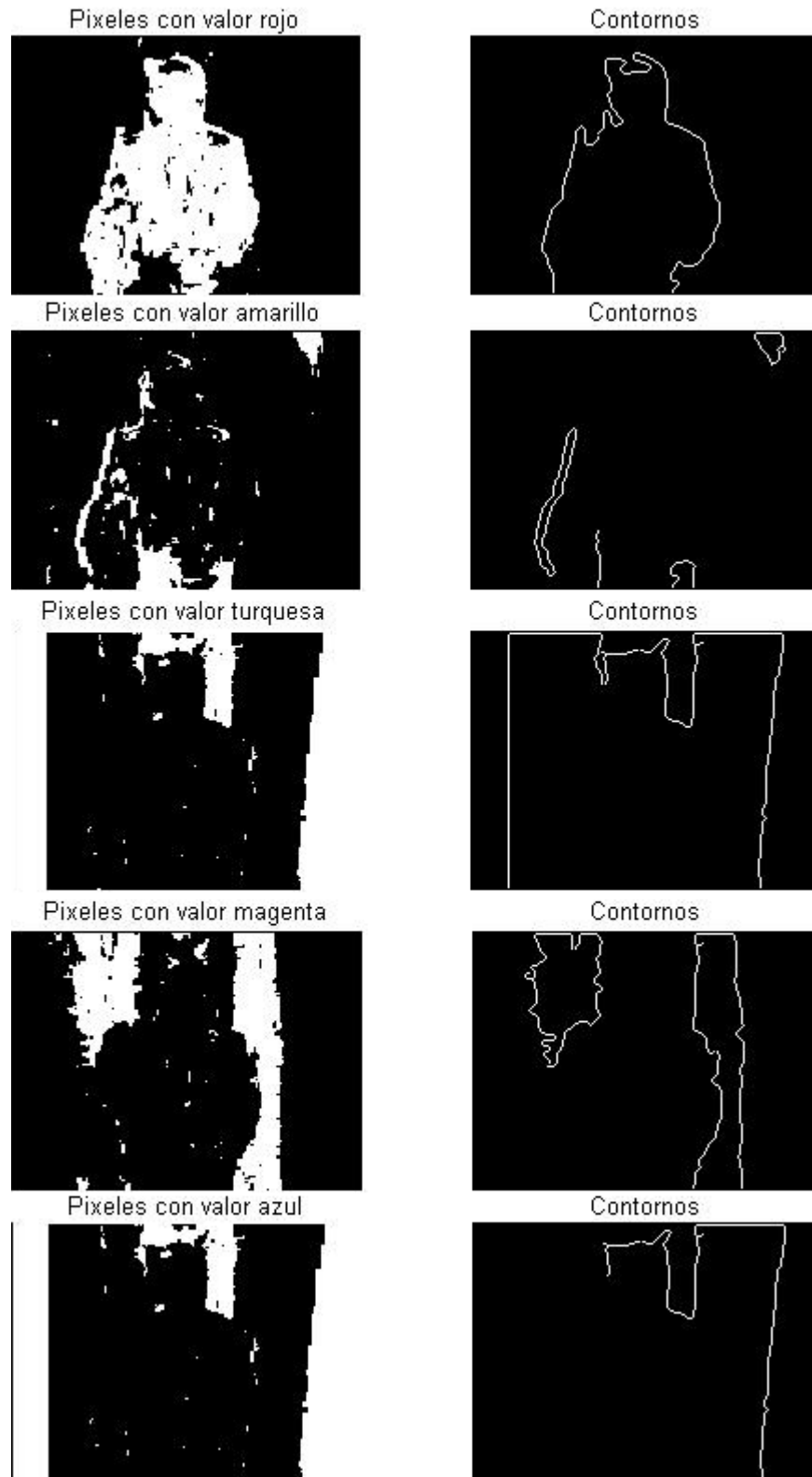


Figura 6.31. Descomposición del mapa de profundidad en función del color y cálculo de los contornos.

Una vez obtenido los valores de contorno anteriores para cada color tan solo queda añadir en la imagen final estos contornos asignándoles un tono y un grosor diferente en función de la distancia, quedando como resultado la imagen de la Figura 6.23 que se muestra en el apartado anterior.

6.2.6.2. Cálculo de contornos usando imagen original

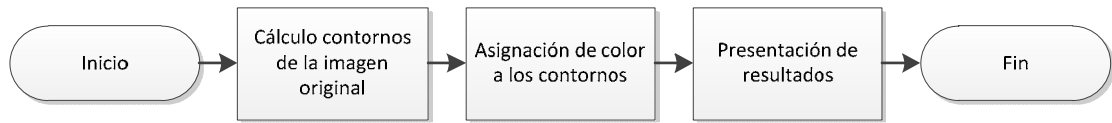


Figura 6.32. Diagrama de flujo del cálculo de contornos usando imagen original.

A diferencia del método anterior, en este caso el cálculo de los contornos se realiza de la imagen original del par estéreo. Esto supone una mayor dificultad a la hora de asignarles color. Lo cual se debe a que los contornos calculados para un mismo objeto tendrán píxeles que corresponderán con un valor del mapa de profundidad de un objeto y con valores de objetos vecinos. Esto sucede porque los contornos calculados con la función *edge* representan la frontera entre los objetos existentes en la imagen.

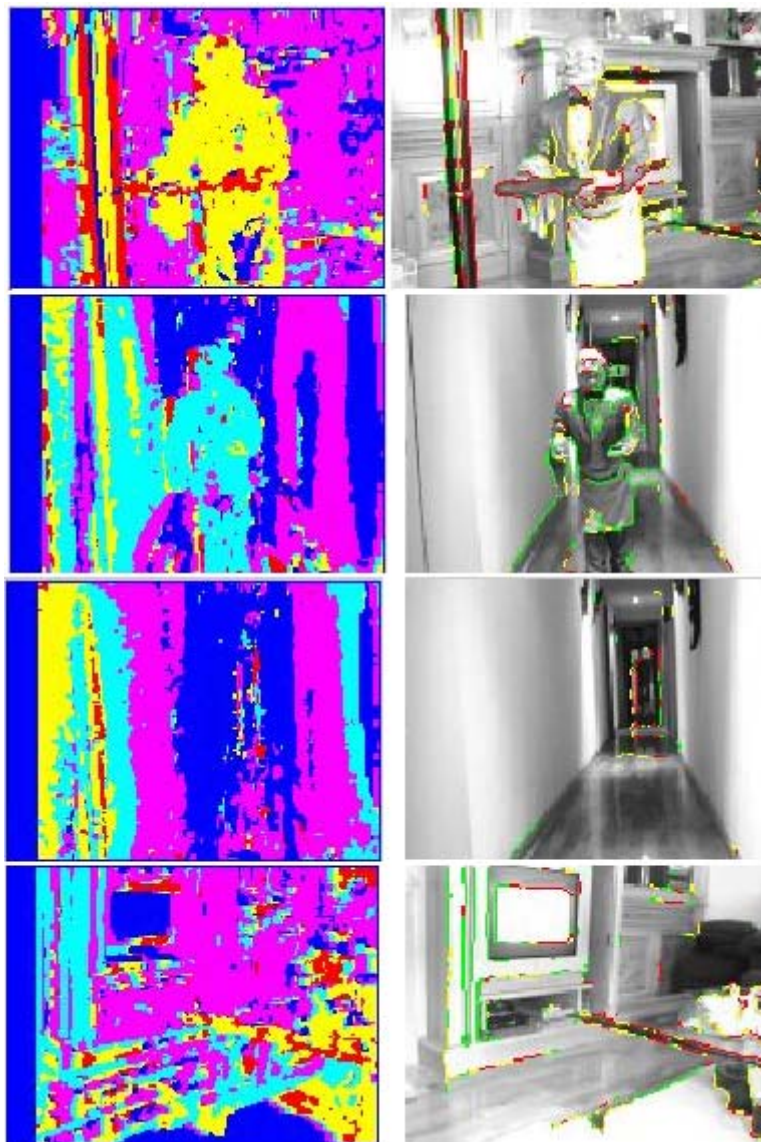


Figura 6.33. Ejemplos del cálculo de contornos usando imagen original.

CAPÍTULO 7. CONCLUSIONES

7.1. Resultados y conclusiones

En este apartado se presentan los resultados obtenidos tras la aplicación de los algoritmos descritos en el apartado anterior. Los resultados mostrados a continuación se han obtenido tras la aplicación de los algoritmos descritos anteriormente sobre un par estéreo captado con la cámara web Minoru 3D, con resolución de 120x160 píxeles. En los siguientes resultados se muestran el efecto de usar diferentes valores de las variables empleadas en el cálculo de la correlación (barrido y tamaño de vecindad) y diferentes tamaños de imagen.

En los resultados se representan tanto los correspondientes mapas de profundidad a color de la escena captada, como el resultado final de aplicar todos los algoritmos desarrollados. Además se indica el valor de barrido y tamaño de la ventana vecindad empleados.

Barrido 10 - Tamaño de la ventana vecindad 3x3



Figura 7.1. Resultado para barrido 10 y vecindad 3x3.

Barrido 10 - Tamaño de la ventana vecindad 5x5

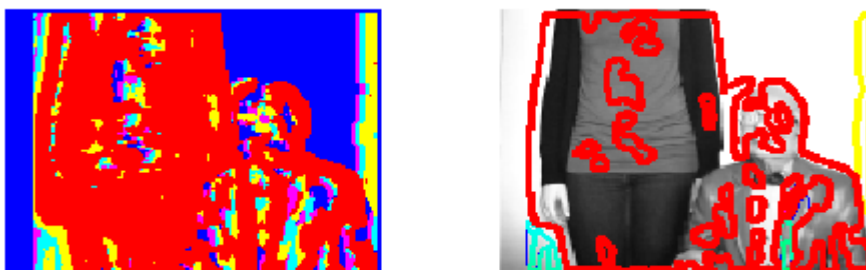


Figura 7.2. Resultado para barrido 10 y vecindad 5x5.

Barrido 10 - Tamaño de la ventana vecindad 7x7

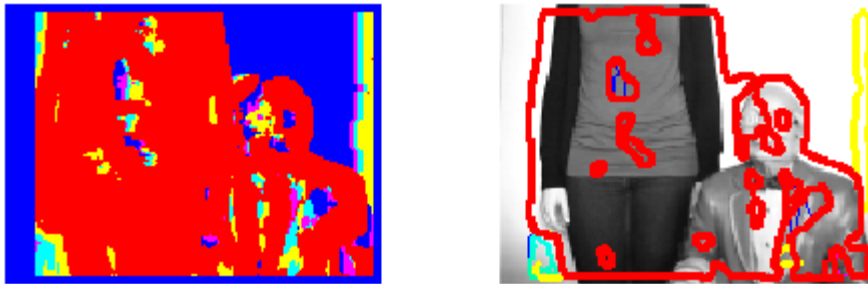


Figura 7.3. Resultado para barrido 10 y vecindad 7x7.

Barrido 13 - Tamaño de ventana vecindad 3x3

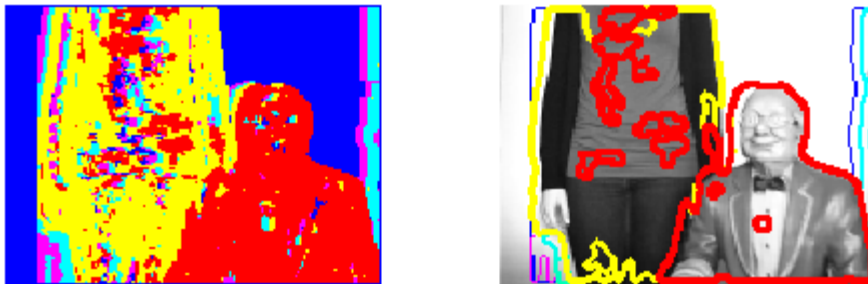


Figura 7.4. Resultado para barrido 13 y vecindad 3x3.

Barrido 13 - Tamaño de la ventana vecindad 5x5

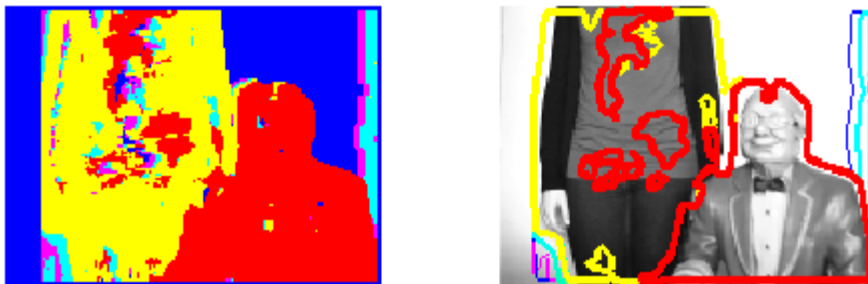


Figura 7.5. Resultado para barrido 13 y vecindad 5x5.

Barrido 13 - Tamaño de la ventana vecindad 7x7



Figura 7.6. Resultado para barrido 13 y vecindad 7x7.

Barrido 15 - Tamaño de la ventana vecindad 3x3

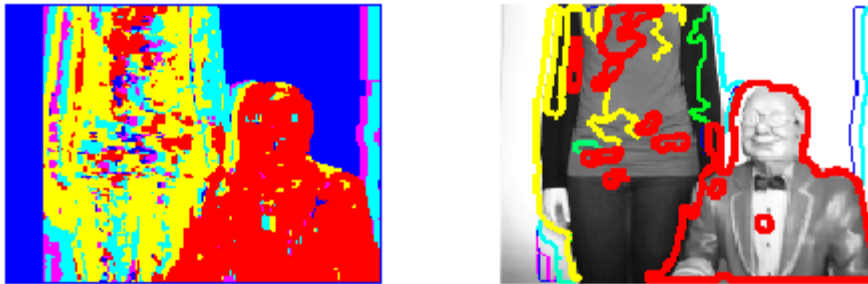


Figura 7.7. Resultado para barrido 15 y vecindad 3x3.

Barrido 15 - Tamaño de la ventana vecindad 5x5

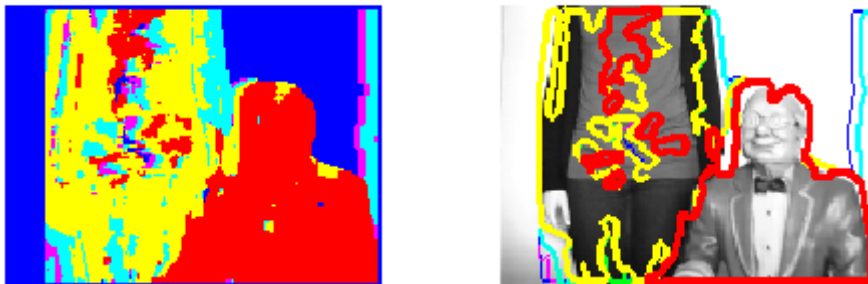


Figura 7.8. Resultado para barrido 15 y vecindad 5x5.

Barrido 15 - Tamaño de la ventana vecindad 7x7

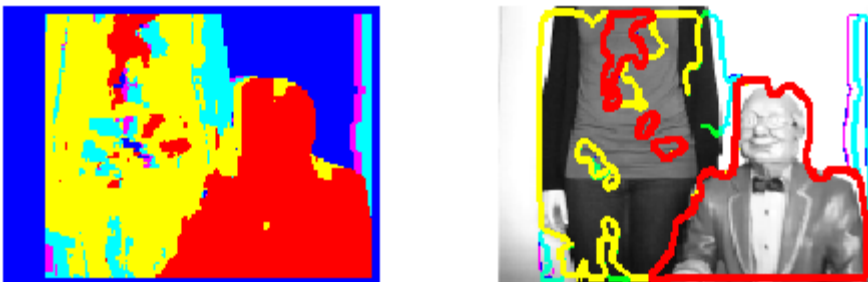


Figura 7.9. Resultado para barrido 15 y vecindad 7x7.

Barrido 20 - Tamaño de la ventana vecindad 3x3

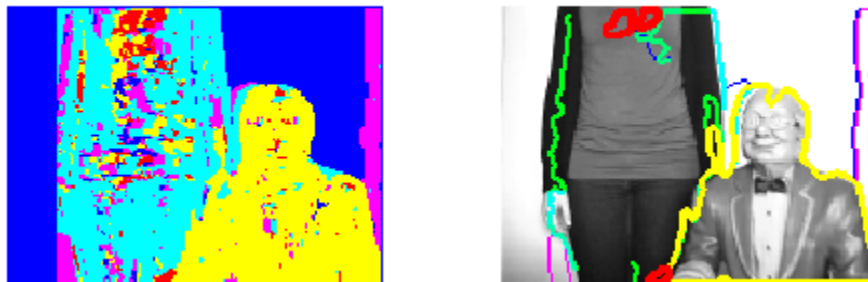


Figura 7.10. Resultado para barrido 20 y vecindad 3x3.

Barrido 20 - Tamaño de la ventana vecindad 5x5

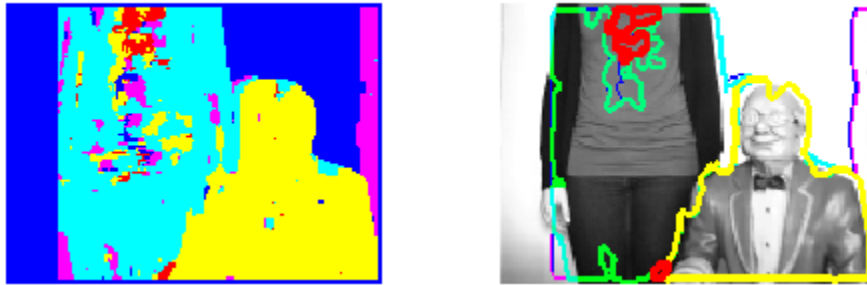


Figura 7.11. Resultado para barrido 20 y vecindad 5x5.

Barrido 20 - Tamaño de la ventana vecindad 7x7

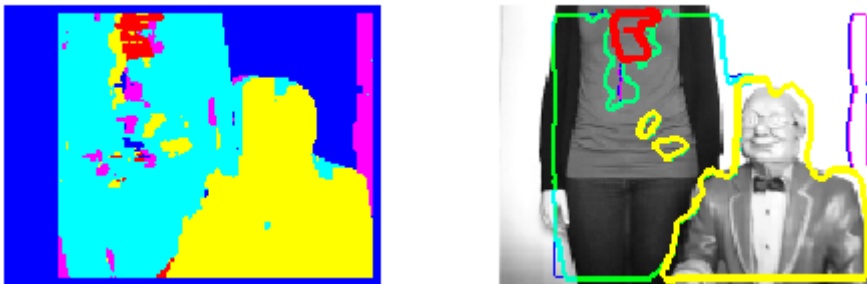


Figura 7.12. Resultado para barrido 20 y vecindad 7x7.

Barrido 25 - Tamaño de la ventana vecindad 3x3

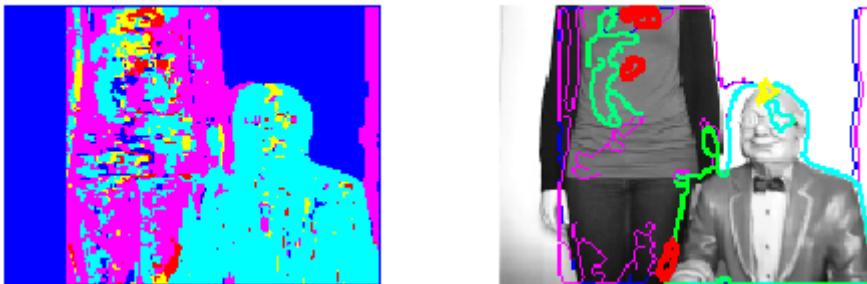


Figura 7.13. Resultado para barrido 25 y vecindad 3x3.

Barrido 25 - Tamaño de la ventana vecindad 5x5

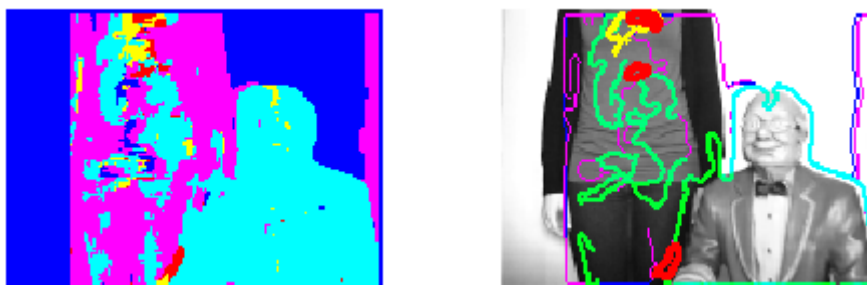


Figura 7.14. Resultado para barrido 25 y vecindad 5x5.

Barrido 25 - Tamaño de la ventana vecindad 7x7

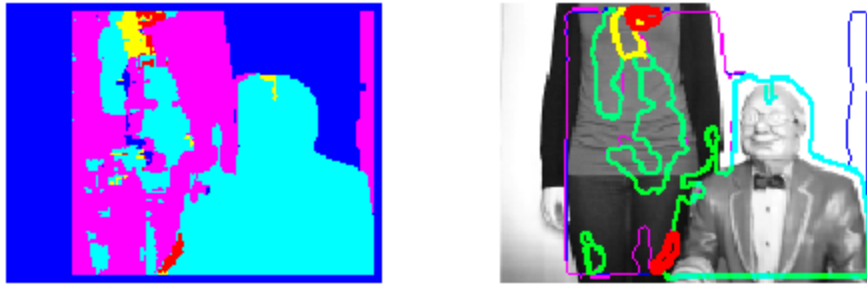


Figura 7.15. Resultado para barrido 25 y vecindad 7x7.

Barrido 30 - Tamaño de la ventana vecindad 3x3

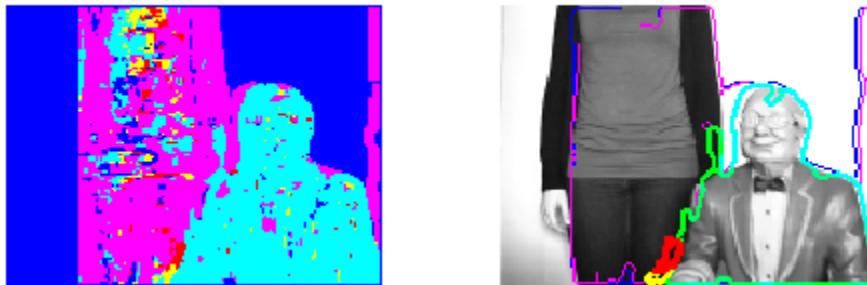


Figura 7.16. Resultado para barrido 30 y vecindad 3x3.

Barrido 30 - Tamaño de la ventana vecindad 5x5



Figura 7.17. Resultado para barrido 30 y vecindad 5x5.

Barrido 30 - Tamaño de la ventana vecindad 7x7



Figura 7.18. Resultado para barrido 30 y vecindad 7x7.

De los resultados mostrados anteriormente se aprecian diferencias sustanciales al usar diferentes valores de las variables barrido y tamaño de la ventana vecindad.

El aumento de la variable barrido tiene como consecuencia, en el resultado final, el aumento del rango de la distancia a la que se encuentra el objeto y la obtención de un mapa de profundidad con menor error.

En cuanto al aumento de la variable que indica el tamaño de la ventana vecindad se observa que implica la obtención de un mapa de profundidad donde los objetos se definen con mayor claridad. Por consiguiente, se obtiene un resultado final en el que los contornos están más ajustados al contorno real de la imagen original.

Así, las variables barrido y tamaño de vecindad deben de llegar a un compromiso en función de la cantidad de información presente en el entorno, distancia en la que se quieren aplicar los algoritmos o tamaño de la imagen original entre otros factores.

Los tiempos de cómputo se ven influenciados notablemente al variar estas variables. A continuación se muestra la variación de los tiempos en función de la variación de los parámetros barrido, ventana vecindad y tamaño de la imagen.

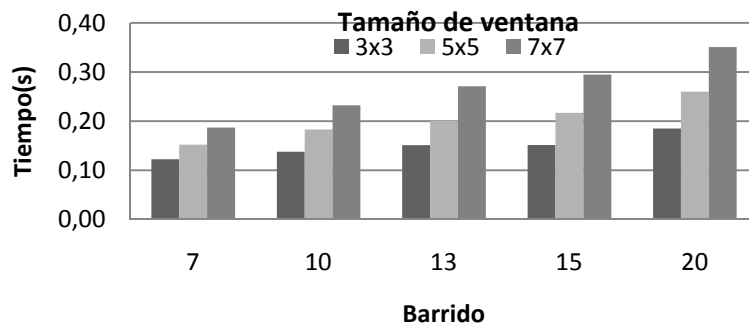


Figura 7.19. Tiempo cómputo para imágenes de 80x105.

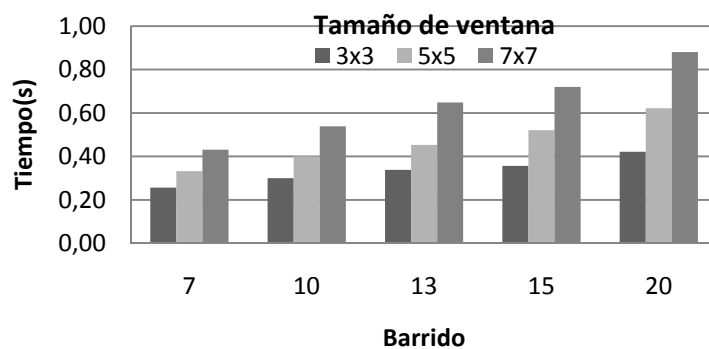


Figura 7.20. Tiempo de cómputo para imágenes de 120x160.

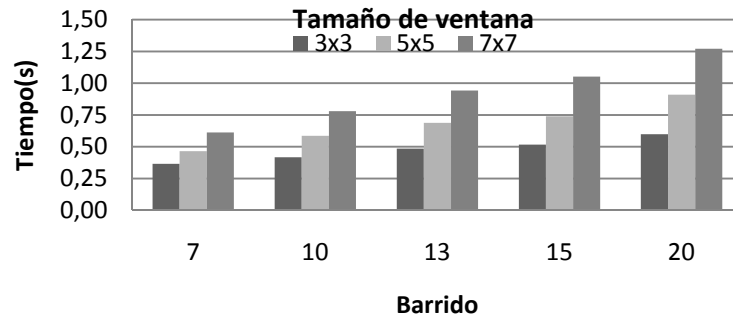


Figura 7.21. Tiempo de cómputo para imágenes de 144x192.

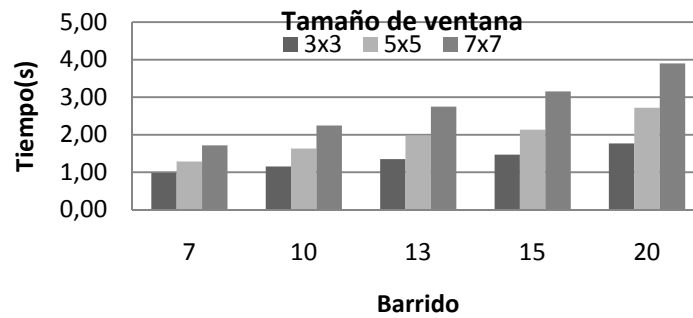


Figura 7.22. Tiempos de cómputo para imágenes de 240x320.

Prestando atención a los resultados anteriores y a los tiempos de cómputo necesarios para la obtención de estos resultados, se aprecia que el objetivo marcado inicialmente con este proyecto, el cual buscaba la adición de información de la distancia de los objetos en su contorno, se ha alcanzado satisfactoriamente. Sin embargo, es necesaria la realización de un estudio clínico con posibles usuarios de este sistema para conocer así la efectividad real del sistema cuando es usado por una persona con problemas visuales.

7.2. Trabajos futuros

El principal problema que presenta la aplicación desarrollada, está relacionada con los tiempos de cómputo necesarios para la ejecución del algoritmo de correlación para el cálculo de distancias, por lo que sería interesante conseguir reducir el coste computacional de este algoritmo lo máximo posible.

El método actual de cálculo de la profundidad aplica el algoritmo de correlación sobre casi todos los píxeles de la imagen, obteniéndose por tanto la distancia de todos los píxeles que conforman la imagen. Sin embargo, de todos estos valores de profundidad calculados, tan solo son utilizados posteriormente una pequeña parte, en concreto los correspondientes a los píxeles de los contornos de los objetos. De esta manera, si tan solo se calculasen los valores de profundidad correspondientes a los contornos de los objetos, el coste computacional del algoritmo se vería reducido notablemente.

CAPÍTULO 8. PRESUPUESTO

El proyecto se puede dividir en tres fases de trabajo fundamentales. La primera consiste en el proceso de análisis de trabajos previos que seguían líneas comunes con este proyecto, estado del arte y tratamiento de imágenes por ordenador.

En la fase número 2 se realiza el desarrollo de un sistema capaz de alcanzar los objetivos buscados. Este bloque incluye el desarrollo de algoritmos y búsqueda de sistemas hardware válidos para implementarlos, además de realizarse las correspondientes pruebas de funcionamiento. La realimentación de resultados es importante para la obtención de un sistema lo más óptimo posible, por lo que la realización de pruebas y desarrollo del sistema se realizó paralelamente en el tiempo.

El último paso consiste en la realización de pruebas del sistema final desarrollado y el desarrollo de la memoria explicativa del trabajo realizado.

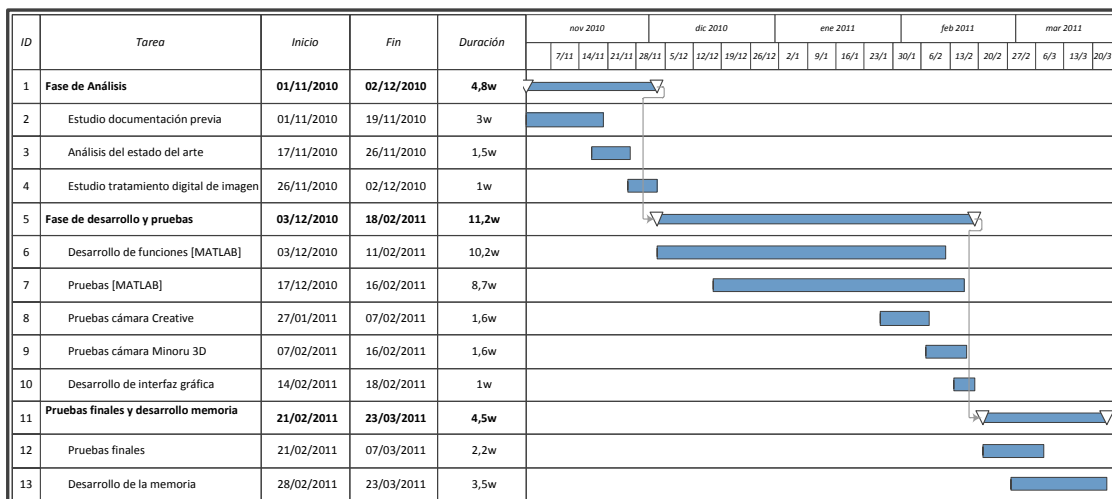


Figura 8.1. Diagrama de Gantt del proyecto.



PRESUPUESTO DE PROYECTO

1.- Autor:

Barranco García, Carlos Antonio

2.- Departamento:

Tecnología Electrónica

3.- Descripción del Proyecto:

- Título: Diseño e implementación de algoritmos de tratamiento de imágenes para ayudas técnicas visuales usando HMDs
- Duración (meses): 5
- Tasa de costes indirectos: 20%

4.- Presupuesto total del Proyecto (valores en Euros):

Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación (hombres mes) ^{*)}	Coste hombre mes	Coste (Euro)	Firma de conformidad
Barranco García, Carlos A.		Ingeniero	5	2.694,39	13.471,95	
Hombres mes 5				Total	13.471,95	

^{*)} 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{*)}
Netbook SAMSUNG	217,00	90	4,5	60	14,65
HMD Trivision ArVision3D	11.650,00	20	1,0	60	38,83
Webcam Minoru 3D	80,00	60	3,0	60	2,40
Notebook HP 14,5" Envy i7	1.599,00	20	1,0	60	5,33
Adaptador HDMI a VGI	5,90	20	1,0	60	0,02
Webcam Creative Live!	38,00	20	1,0	60	0,13
Total					61,38

^{*)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = n° de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
Total		0,00

OTROS COSTES DIRECTOS DEL PROYECTO^{*)}

Descripción	Empresa	Costes imputable
Total		0,00

^{*)} Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	13.472
Amortización	61
Subcontratación de tareas	0
Costes de funcionamiento	0
Costes indirectos	2.707
Total	16.240

El presupuesto total de este proyecto asciende a la cantidad de 16.240€.

Leganés a 14 de Abril de 2011

El ingeniero proyectista

Fdo. Carlos A. Barranco García

REFERENCIAS

[1] Collado Martínez, F.J. “Sistema de ayuda a la movilidad para personas con baja visión”, UC3M, 2008

[2] Girón Sáenz, R. “Diseño e implementación de un sistema estereoscópico como ayuda técnica para Baja Visión”, UC3M, Enero 2010

[3] Organización Mundial de la Salud (OMS) [Internet]: <<http://www.who.int/>> [4] Wikipedia [Internet]: <<http://es.wikipedia.org/>>

[5] Instituto de Oftalmología Aplicada IOBA [Internet]: <<http://www.ioba.med.uva.es/>>

[6] ONCE [Internet]: <<http://www.once.es/>>

[7] Accesibles.org [Internet]: <<http://accesibles.org/>>

[8] [Internet]: <<http://oftall.mejorforo.net/>>

[9] Sociedad Española de Especialistas en Baja Visión [Internet]: <<http://www.seebv.com/>>

[10] Albinism [Internet]: <<http://www.albinism.org/>>

[11] Discapnet [Internet]: <<http://salud.discapnet.es/>>

[12] Ocularis [Internet]: <<http://ocularis.es/>>

[13] Vision 3D [Internet]: <<http://www.vision3d.com/stereo.html>>

[14] Realidad Virtual [Internet]: <<http://www.realidadvirtual.com/>>

[15] R. T. Azuma, “A survey of Augmented Reality”, Hughes Research Laboratories, Malibu CA 90265, [Internet]: <<http://www.cs.unc.edu/~azuma/>>

[16] VrLogic[Internet]: <http://www.vrlogic.com/html/head_mounted_displays.html>

[17] James E. Melzer, Kirk Moffitt “Head Mounted Displays. Designing for the User”, Mc. Graw-Hill.

- [18] Trivisio [Internet]: <<http://www.trivisio.com/>>
- [19] Xbox [Internet]: <<http://www.xbox.com/es-es/kinect>>
- [20] Creative [Internet]: <<http://es.creative.com/>>
- [21] Minoru 3D [Internet]: <<http://www.minoru3d.com/>>
- [22] Zivora [Internet]: <<http://www.zivora.es/>>
- [23] Fujifilm [Internet]: <<http://www.fujifilm.es/>>
- [24] Inlife [Internet]: <<http://www.3dinlife.com/en/download.html>>
- [25] Mathworks [Internet]: <<http://www.mathworks.com/>>
- [26] Y. Nakamura, T. Matsuura, K. Satoh, y Y. Ohta. "Occlusion detectable stereo—occlusion patterns in camera matrix." en *CVPR*, páginas 371–378, 1996.
- [27] Gonzales, R. C., Woods, R.E., "Digital Image Processing", Addison-Wesley, Reading, MA, 1992.
- [28] Takeo Kanade, Hiroshi Kano, Shigeru Kimura, Atsushi Yoshida, Kazuo Oda, "Development of a Video-Rate Stereo Machine" en *Proceedings of International Robotics and Systems Conference (IROS'95)*, Pittsburgh PA , August 5-9, 1995.
- [29] González Velázquez, D., "Implementación de un Sistema de Visión Estéreo Mediante Instrucciones SIMD", UC3M, Diciembre, 2006.

AXEXO 1. ALGORITMOS DESARROLLADOS

El objetivo de este anexo es el de dar a conocer al lector los algoritmos que se han desarrollado para alcanzar el objetivo que se había marcado con la realización de este trabajo.

A continuación se muestran los algoritmos implementados para la adquisición de imágenes, calibración de los sistemas de adquisición, preprocesamiento de imagen, cálculo de profundidades y asignación de contornos de diferentes colores a los objetos en función de su distancia.

El cálculo de los contornos o de los mapas de disparidad se ha realizado con la implementación de diferentes algoritmos, ya que en función del tipo de discapacidad o del tipo de uso que el paciente vaya a hacer de la herramienta desarrollada, puede ser más interesante una opción u otra.

Los mapas de profundidad de los pares estereoscópicos se han calculado siempre siguiendo en método de la correlación de la suma de diferencias absolutas, SAD, como se explicó en el capítulo sexto. Sin embargo, también se implementaron los algoritmos para el cálculo de la correlación usando los métodos ZSAD, ZNSSD y NSSD, por lo que también se muestran a continuación.

En cuanto al cálculo de contornos, como se describió en el capítulo sexto, se ha realizado o bien utilizando la imagen original o el mapa de profundidad. Además, conviene recordar que la asignación de la información relativa a la distancia se había hecho cambiando los colores de los contornos, cambiando su espesor o haciendo cambiando color y espesor.

Los algoritmos que se muestran hacen llamadas a las funciones *calibrado_fourier.m* y *correlación_sad.m* los cuales se muestran al final de este anexo.

- **Mapa de profundidades a color (usando imágenes tomadas previamente).**

```

% Función principal para el cálculo de mapas de profundidades usando
el método de la suma de diferencias absolutas, sin aplicar previamente
un preprocesado.
% Cálculo para un par estéreo tomado previamente (no modo stream).
% Representación de mapa de profundidad a color.
% Contornos varían de color y de grosor.

% Lectura de las imágenes.
imagen_right = imread('right.png');
imagen_left = imread('left.png');
% Redimensionamiento de las imágenes.
imagen_right=imresize(imagen_right,[144,192]);
imagen_left=imresize(imagen_left,[144,192]);
% Paso de RGB a escala de grises.
imagen_lg=rgb2gray(imagen_left); imagen_rg=rgb2gray(imagen_right);
% Cambio de datos a tipo double.
imagen_l=im2double(imagen_lg); imagen_r=im2double(imagen_rg);
% Matrices con los valores R,G,B (usada posteriormente para añadir
bordes a color).
imR=double(imagen_rg)./255;imG=double(imagen_rg)./255;
imB=double(imagen_rg)./255;
% Lectura e inicialización de valores para cálculo de la correlación.
n=input('Introducir valor de barrido: ','s');
nP=str2num(n);
ventana=input('Introducir valor de ventana(1,2,3,..): ','s');
M=str2num(ventana);
[nrLeft ncLeft]=size(imagen_l); [nrRight ncRight]=size(imagen_r);
dispMap=zeros(nrLeft,ncLeft);
% Cálculo de la correlación.
tic;
for i=1+M:nrRight-M
    for j=1+M+nP:ncRight-M
        Crs=correlacion_sad(M,i,j,imagen_r,imagen_l,nP);
        [k,dispMap(i,j)]=min(Crs);
    end
end
tiempo_correlacion=toc;
% Cálculo del mapa de profundidades a color.
tic;
dispMap=dispMap/nP;
dispMapR=zeros(nrLeft,ncLeft);dispMapG=zeros(nrLeft,ncLeft);
dispMapB=zeros(nrLeft,ncLeft);
for i=1:nrLeft
    for j=1:ncLeft
        if dispMap(i,j)>=0.8
            dispMapR(i,j)=1; dispMapG(i,j)=0; dispMapB(i,j)=0;
        elseif dispMap(i,j)>=0.6
            dispMapR(i,j)=1; dispMapG(i,j)=1; dispMapB(i,j)=0;
        elseif dispMap(i,j)>=0.4
            dispMapR(i,j)=0; dispMapG(i,j)=1; dispMapB(i,j)=1;
        elseif dispMap(i,j)>=0.2
            dispMapR(i,j)=1; dispMapG(i,j)=0; dispMapB(i,j)=1;
        else
            dispMapR(i,j)=0; dispMapG(i,j)=0; dispMapB(i,j)=1;
        end
    end
end
end
imdispTotal=cat(3,dispMapR,dispMapG,dispMapB);
tiempo_color=toc

```

```

% Cálculo de contornos mapa de profundidades en color y con difentes
% grosores. La matriz dilate indica el grosor del contorno.
tic;
rango1=im2double(dispmat>=0.8);
contorno1=edge(rango1,'canny',0.9);
dilate=ones(3,3);
contorno1=imdilate(contorno1,dilate);
rango2=im2double(dispmat>=0.6&dispmat<0.8);
contorno2=edge(rango2,'canny',0.99);
dilate=[0 1 0; 1 1 1; 0 1 0];
contorno2=imdilate(contorno2,dilate);
rango3=im2double(dispmat>=0.4&dispmat<0.6);
contorno3=edge(rango3,'canny',0.99);
dilate=[0 0 0; 0 1 1; 0 1 1];
contorno3=imdilate(contorno3,dilate);
rango4=im2double(dispmat>=0.2&dispmat<0.4);
contorno4=edge(rango4,'canny',0.99);
dilate=[0 0 0; 1 1 1; 0 0 0];
contorno4=imdilate(contorno4,dilate);
rango5=im2double(dispmat<0.6&dispmat>0.1);
contorno5=edge(rango5,'canny',0.99);

for i=1:nrLeft
    for j=1:ncLeft
        if ((contorno1(i,j)==1))
            imR(i,j)=1;imG(i,j)=0;imB(i,j)=0;
        elseif (contorno2(i,j)==1)
            imR(i,j)=1;imG(i,j)=1;imB(i,j)=0;
        elseif (contorno3(i,j)==1)
            imR(i,j)=0;imG(i,j)=1;imB(i,j)=1;
        elseif (contorno4(i,j)==1)
            imR(i,j)=1;imG(i,j)=0;imB(i,j)=1;
        elseif (contorno5(i,j)==1)
            imR(i,j)=0;imG(i,j)=0;imB(i,j)=1;
        end
    end
end

imTotal=cat(3,imR,imG,imB);
tiempo_bordes=toc;
tiempo=tiempo_color+tiempo_correlacion+tiempo_bordes;
% Representación del mapa de profundidad de la imagen
subplot(1,2,1),imshow(imdispTotal)
title(['Mapa de profundidades'],['Tiempo: ', num2str(tiempo), ' s'])
% Representación de las imágenes con contornos.
subplot(1,2,2),imshow(imTotal)
title(['Barrido: ',int2str(nP)],['Ventanas de: ', int2str(2*M+1),
'x',int2str(2*M+1)])

```


- **Cálculo de bordes haciendo uso del mapa de profundidad.**

```

% Función que calcula los bordes de las imágenes tomadas por las
cámaras, haciendo uso del mapa de profundidad (a color).
% Se realiza una calibración previa de las cámaras que componen el
sistema de adquisición.
% Ver capítulo 6.2.6.1.

% Borrar variables, limpiar pantalla y cerrar objetos de video.
imagreset;close all; clear all;
% Creación e inicialización del objeto de video 1.
vid_l=videoinput('winvideo',1,'RGB24_160x120');
preview(vid_l);
start(vid_l);
% Creación e inicialización del objeto de vídeo 2.
vid_r=videoinput('winvideo',2,'RGB24_160x120');
preview(vid_r);
start(vid_r);
% Enfoque de las camaras a la plantilla de calibración.
a=input('Enfoca las camaras sobre la plantilla de calibracion y
pulsa enter','s');
% Captura de la primera imagen (usada para calibración).
imagen_right=getsnapshot(vid_r);
gray_right=rgb2gray(imagen_right);
imagen_left=getsnapshot(vid_l);
gray_left=rgb2gray(imagen_left);
% Inicializaciones
[n m]=size(gray_right);
pos_r=0;pos_l=0;

% Cálculo de la calibración.
while (pos_r~=pos_l) || (pos_l==0)
    % Captura de imágenes y paso de RGB a niveles de gris.
    imagen_right=getsnapshot(vid_r);
    gray_right=rgb2gray(imagen_right);
    imagen_left=getsnapshot(vid_l);
    gray_left=rgb2gray(imagen_left);
    dif=pos_r-pos_l;
    if (pos_r>pos_l)
        % Nueva dimensión de la matriz right y left.
        gray_right_calib=zeros(n-(dif),m);
        gray_left_calib=zeros(n-(dif),m);
        % Nuevos valores para right y left.
        gray_right_calib(1:n-(dif),:)=gray_right(1+(dif):n,:);
        gray_left_calib(:,:)=gray_left(1:n-(dif),:);
    elseif (pos_l>pos_r)
        gray_right_calib=zeros(n+dif,m);
        gray_left_calib=zeros(n+dif,m);
        gray_left_calib(1:n+dif,:)=gray_left(1-dif:n,:);
        gray_right_calib(:,:)=gray_right(1:n+dif,:);
    end
    if (pos_l==0)
        [pos_r pos_l]=calibrado_fourier(gray_right, gray_left)
    else
        [pos_r pos_l]=calibrado_fourier(gray_right_calib,
gray_left_calib)
    end
end

% Inicializaciones
[nrLeft ncLeft]=size(gray_left_calib);

```

```

    [nrRight ncRight]=size(gray_right_calib);
    dispMap=zeros(nrLeft,ncLeft);
% Lectura de valores necesarios para el calculo de la correlación.
    barrido=input('Introducir valor de barrido: ','s');
    nP=str2num(barrido);
    ventana=input('Introducir valor de la ventana(1,2,3,...): ','s');
    M=str2num(ventana);
% Bucle para cálculo de correlación y presentación de resultados.
while(1)
    imagen_left=getsnapshot(vid_l);
    imagen_right=getsnapshot(vid_r);
    gray_right=rgb2gray(imagen_right);
    gray_left=rgb2gray(imagen_left);
    if dif>0
        gray_right_calib(:,:)=gray_right((1+dif):n,:);
        gray_left_calib(:,:)=gray_left(1:(n-dif),:);
    else
        gray_left_calib(:,:)=gray_left((1-dif):n,:);
        gray_right_calib(:,:)=gray_right(1:(n+dif),:);
    end
    imR=double(gray_left_calib)./255;
    imG=double(gray_left_calib)./255;
    imB=double(gray_left_calib)./255;
% Cálculo de la correlacion SAD.
    tic;
    imagen_r=im2double(gray_right_calib);
    imagen_l=im2double(gray_left_calib);
    for i=1+M:nrRight-M
        for j=1+M+nP:ncRight-M
            Crs=correlacion_sad(M,i,j,imagen_l,imagen_r,nP);
            [k,dispMap(i,j)]=min(Crs);
        end
    end
    tiempo=toc;
% Mapas de profundidades en color.
    dispMapR=zeros(nrLeft,ncLeft);dispMapG=zeros(nrLeft,ncLeft);
    dispMapB=zeros(nrLeft,ncLeft);
    for i=1:nrLeft
        for j=1:ncLeft
            if dispMap(i,j)>=0.8
                dispMapR(i,j)=1; dispMapG(i,j)=0; dispMapB(i,j)=0;
            elseif dispMap(i,j)>=0.6
                dispMapR(i,j)=1; dispMapG(i,j)=1; dispMapB(i,j)=0;
            elseif dispMap(i,j)>=0.4
                dispMapR(i,j)=0; dispMapG(i,j)=1; dispMapB(i,j)=1;
            elseif dispMap(i,j)>=0.2
                dispMapR(i,j)=1; dispMapG(i,j)=0; dispMapB(i,j)=1;
            else
                dispMapR(i,j)=0; dispMapG(i,j)=0; dispMapB(i,j)=1;
            end
        end
    end
    imdispTotal=cat(3,dispMapR,dispMapG,dispMapB);
    subplot(1,2,1),imshow(imdispTotal)
    title(['Mapa de profundidades'], ['Tiempo: ', num2str(tiempo), '
s']})
    tiempo_color=toc;
    tic;
    dilate=ones(3,3);
    rangol=im2double(dispMap>=0.8);
    contornol=edge(rangol,'canny',0.9);

```

```

contorno1=imdilate(contorno1,dilate);
rango2=im2double(dispmat>=0.6&dispmat<0.8);
contorno2=edge(rango2,'canny',0.99);
contorno2=imdilate(contorno2,dilate);
rango3=im2double(dispmat>=0.4&dispmat<0.6);
contorno3=edge(rango3,'canny',0.99);
rango4=im2double(dispmat>=0.2&dispmat<0.4);
contorno4=edge(rango4,'canny',0.99);
rango5=im2double(dispmat<0.6&dispmat>0.1);
contorno5=edge(rango5,'canny',0.99);
for i=1:nrLeft
    for j=1:ncLeft
        if (contorno1(i,j)==1)
            imR(i,j)=1;imG(i,j)=0;imB(i,j)=0;
        elseif (contorno2(i,j)==1)
            imR(i,j)=1;imG(i,j)=1;imB(i,j)=0;
        elseif (contorno3(i,j)==1)
            imR(i,j)=0;imG(i,j)=1;imB(i,j)=1;
        elseif (contorno4(i,j)==1)
            imR(i,j)=1;imG(i,j)=0;imB(i,j)=1;
        elseif (contorno5(i,j)==1)
            imR(i,j)=0;imG(i,j)=0;imB(i,j)=1;
        end
    end
end

imTotal=cat(3,imR,imG,imB);
tiempo_bordes=toc;
% Representación las imágenes más los contornos.
subplot(1,2,2),imshow(imTotal)
title({'Barrido: ',int2str(nP)},{ 'Ventanas de: ',
int2str(2*M+1),'x',int2str(2*M+1)]]}
end

```

- Cálculo de bordes haciendo uso de la imagen original.

```

% Función que calcula los contornos de las imágenes tomadas por las
cámaras, haciendo uso de la imagen original.
% Se realiza una calibración previa de las cámaras que componen el
sistema de adquisición.
% Se recomienda ver capítulo 6.2.6.2.

imaqreset;close all; clear all;
% Inicialización objeto de video.
vid_l=videoinput('winvideo',1,'RGB24_160x120');
preview(vid_l);start(vid_l);
% Creación e inicialización del objeto de vídeo 2 .
vid_r=videoinput('winvideo',2,'RGB24_160x120');
preview(vid_r);start(vid_r);
% Captura de la primera imagen.
imagen_right=getsnapshot(vid_r);
gray_right=rgb2gray(imagen_right);
imagen_left=getsnapshot(vid_l);
gray_left=rgb2gray(imagen_left);
% Inicializaciones.
[n m]=size(gray_right); pos_r=0;pos_l=0;
a=input('Enfoca las camaras sobre la plantilla de calibracion y
pulsa enter','s');
% Calibración.

```

```

while (pos_r~=pos_l) || (pos_l==0)
    imagen_right=getsnapshot(vid_r);
    gray_right=rgb2gray(imagen_right);
    imagen_left=getsnapshot(vid_l);
    gray_left=rgb2gray(imagen_left);
    dif=pos_r-pos_l;
    if (pos_r>pos_l)
        gray_right_calib=zeros(n-(dif),m);
        gray_left_calib=zeros(n-(dif),m);
        gray_right_calib(1:n-(dif),:)=gray_right(1+(dif):n,:);
        gray_left_calib(:,:)=gray_left(1:n-(dif),:);
    elseif (pos_l>pos_r)
        gray_right_calib=zeros(n+dif,m);
        gray_left_calib=zeros(n+dif,m);
        gray_left_calib(1:n+dif,:)=gray_left(1-dif:n,:);
        gray_right_calib(:,:)=gray_right(1:n+dif,:);
    end
    if (pos_l==0)
        [pos_r pos_l]=calibrado_fourier(gray_right, gray_left)
    else
        [pos_r pos_l]=calibrado_fourier(gray_right_calib,
gray_left_calib)
    end
end

[nrLeft ncLeft]=size(gray_left_calib);
[nrRight ncRight]=size(gray_right_calib);
dispMap=zeros(nrLeft,ncLeft);
% Introduccion de valores
barrido=input('Introducir valor de barrido: ','s');
nP=str2num(barrido);
ventana=input('Introducir valor de ventana (1,2,3,..): ','s');
M=str2num(ventana); N=M;
valor_g=input('Introducir número de grises: ','s');
ngrises=str2num(valor_g);
while(1)
    imagen_left=getsnapshot(vid_l);
    imagen_right=getsnapshot(vid_r);
    gray_right=rgb2gray(imagen_right);
    gray_left=rgb2gray(imagen_left);
    if dif>0
        gray_right_calib(:,:)=gray_right((1+dif):n,:);
        gray_left_calib(:,:)=gray_left(1:(n-dif),:);
    else
        gray_left_calib(:,:)=gray_left((1-dif):n,:);
        gray_right_calib(:,:)=gray_right(1:(n+dif),:);
    end
    imR=double(gray_right_calib)./255;
    imG=double(gray_right_calib)./255;
    imB=double(gray_right_calib)./255;
    % Cálculo de la correlación.
    tic;
    imagen_r=im2double(gray_right_calib);
    imagen_l=im2double(gray_left_calib);
    for i=1+M:nrRight-M
        for j=1+M+nP:ncRight-M
            Crs=correlacion_sad(M,i,j,imagen_l,imagen_r,nP);
            [k,dispMap(i,j)]=min(Crs);
        end
    end
    tiempo=toc;

```

```

% Mapa de profundidades de la imagen.
dispMap=dispMap/nP;
dispMapR=zeros(nrLeft,ncLeft);dispMapG=zeros(nrLeft,ncLeft);
dispMapB=zeros(nrLeft,ncLeft);
for i=1:nrLeft
    for j=1:ncLeft
        if dispMap(i,j)>=0.8
            dispMapR(i,j)=1; dispMapG(i,j)=0; dispMapB(i,j)=0;
        elseif dispMap(i,j)>=0.6
            dispMapR(i,j)=1; dispMapG(i,j)=1; dispMapB(i,j)=0;
        elseif dispMap(i,j)>=0.4
            dispMapR(i,j)=0; dispMapG(i,j)=1; dispMapB(i,j)=1;
        elseif dispMap(i,j)>=0.2
            dispMapR(i,j)=1; dispMapG(i,j)=0; dispMapB(i,j)=1;
        else
            dispMapR(i,j)=0; dispMapG(i,j)=0; dispMapB(i,j)=1;
        end
    end
end
end
imdispTotal=cat(3,dispMapR,dispMapG,dispMapB);
subplot(1,2,1),imshow(imdispTotal)
title(['Mapa de profundidades'], ['Tiempo: ', num2str(tiempo), '
s']})
tiempo_color=toc;
tic;
dilate=ones(3,3);
contorno=edge(gray_right_calib, 'canny', 0.3);
rango1=im2double(dispMap>=0.8);
rango1=imdilate(rango1,dilate);
rango2=im2double(dispMap>=0.6&dispMap<0.8);
rango2=imdilate(rango2,dilate);
rango3=im2double(dispMap>=0.4&dispMap<0.6);
rango3=imdilate(rango3,dilate);
for i=1:nrLeft
    for j=1:ncLeft
        if (contorno(i,j)==1)
            if (rango1(i,j)==1)
                imR(i,j)=1;imG(i,j)=0;imB(i,j)=0;
            elseif (rango2(i,j)==1)
                imR(i,j)=1;imG(i,j)=1;imB(i,j)=0;
            elseif (rango3(i,j)==1)
                imR(i,j)=0;imG(i,j)=1;imB(i,j)=0;
            end
        end
    end
end
end
imTotal=cat(3,imR,imG,imB);
tiempo_bordes=toc;
subplot(1,2,2),imshow(imTotal)
title(['Barrido: ',int2str(nP)], ['Ventanas de: ',
int2str(2*M+1), 'x',int2str(2*M+1)], ['y N°gris: ',int2str(ngrises)]})
end

```

- **Calibrado mediante FFT, *calibrado_fourier.m*.**

```

function [pos_r pos_l]=calibrado_fourier(right, left)
% Calibración de las cámaras en el eje vertical, mediante la
transformada rápida de Fourier (FFT) para el según el método visto en

% Lectura de la imagen patrón a buscar en las imágenes captadas por la
cámara(right y left)
cuadrado_RGB=imread('cuadrado.png');
cuadrado=rgb2gray(cuadrado_RGB);
[n m]=size(right);

%Busqueda del cuadrado en las imágenes mediante la FFT.

Fourier_right=real(ifft2(fft2(right).*fft2(rot90(cuadrado,2),n,m)));
fourier_left=real(ifft2(fft2(left).*fft2(rot90(cuadrado,2),n,m)));

% Umbralización del espectro de fouier para la imagen derecha.
Thresh_right=1.*max(fourier_right(:));
final_right=fourier_right/thresh_right;
subplot(2,2,3),imshow(final_right,[])
for i=1:n
    for j=1:m
        if (final_right(i,j)>=0.999985)
            final_right_umb(i,j)=255;
        else
            final_right_umb(i,j)=0;
        end
    end
end
[val pos_r]=max(final_right_umb,[],2);
[val pos_r]=max(val,[],1);
% Umbralización del espectro de fouier para la imagen izquierda.
Thresh_left=1.*max(fourier_left(:));
final_left=fourier_left/thresh_left;
subplot(2,2,1),imshow(final_left,[])
for i=1:n
    for j=1:m
        if (final_left(i,j)>=0.999985)
            final_left_umb(i,j)=255;
        else
            final_left_umb(i,j)=0;
        end
    end
end
[val pos_l]=max(final_left_umb,[],2);
[val pos_l]=max(val,[],1);
% Representación de los resultados.
Figure(3)
subplot(2,2,1),imshow(final_left,[])
subplot(2,2,2),imshow(final_left_umb)
subplot(2,2,3),imshow(final_right,[])
subplot(2,2,4),imshow(final_right_umb)

```

- Preprocesamiento.

El preprocesamiento de la imagen se realiza añadiendo tan solo tres líneas de código a los algoritmos anteriores después de transformar la matriz de la imagen en niveles de gris a tipo *double*. A continuación se muestran el código para realizar diferentes preprocesamientos.

```
% Preprocesado usando la laplaciana.
W=fspecial('laplacian',0);
imagen_r=imfilter(imagen_r,w,'replicate');
imagen_l=imfilter(imagen_l,w,'replicate');

% Preprocesado usando la Gausiana.
W=fspecial('gaussian',5,0.7);
imagen_r=imfilter(imagen_r,w,'replicate');
imagen_l=imfilter(imagen_l,w,'replicate');

% Preprocesado usando la Laplaciana de la gaussiana.
W=fspecial('log',5,0.7);
imagen_r=imfilter(imagen_r,w,'replicate');
imagen_l=imfilter(imagen_l,w,'replicate');
```

- Cálculo de la correlación por el método SAD, *correlacion_sad.m*.

```
function [Cr_vect]=correlacion_sad(ancho,px,py,muestra,imagen,barrido)
% Cálculo de la correlación haciendo uso del método matemático de la
% suma de diferencias absolutas (SAD).
% Ver capítulo 6.2.5.
Cr_vect=zeros(1,barrido,'double');
for h=1:barrido
    for f=px-ancho:1:px+ancho
        for g=py-ancho:1:py+ancho
            Cr_vect(h)=Cr_vect(h)+abs(muestra(f,g)-imagen(f,g-h));
        end
    end
end
end
```

- Cálculo de la correlación por el método ZSAD.

```
function [Cr_vect]=correlacion_zsad(ancho,px,py,muestra,image,barrido)
% Cálculo de la correlación haciendo uso del método matemático de la
% suma de diferencias absolutas con media cero (ZSAD).
% No usado como solución final para el cálculo de la correlación.
% Ver capítulo 6.2.5.

Cr_vect=zeros(1,barrido,'double');num=zeros(1,barrido,'double');
den2=zeros(1,barrido,'double');den=zeros(1,barrido,'double');
den1=0;
% Calculo numerador
for h=1:barrido
    for f=px-ancho:1:px+ancho
        for g=py-ancho:1:py+ancho
            num(h)=num(h)+abs(muestra(f,g)-image(f,g-h));
        end
    end
end
end
% Calculo denominador
```

```

for h=1:barrido
    for f=px-ancho:1:px+ancho
        for g=py-ancho:1:py+ancho
            den2(h)=den2(h)+(image(f,g-h))^2;
        end
    end
end
for f=px-ancho:1:px+ancho
    for g=py-ancho:1:py+ancho
        den1=den1+(muestra(f,g))^2;
    end
end
% Calculo ZSAD
den=den1*den2;
for h=1:barrido
    Cr_vect(h)=num(h)/den(h);
end
end

```

- **Cálculo de la correlación por el método NSSD.**

```

function [Cr_vect]=nssd(ancho,px,py,imagen_left,imagen_right,barrido)
% Cálculo de la correlación mediante el método de la suma normalizada
de diferencias cuadradas NSSD.
% No usado como solución final para el cálculo de la correlación.

cr1=0;Cr_vect=zeros(1,barrido,'double');den1=0; alto=ancho;
    for f=px-ancho:1:px+ancho
        for g=py-alto:1:py+alto
            den1=den1+(imagen_left(f,g))^2;
        end
    end
    den1=sqrt(den1);
for h=1:barrido
    SSD=0;
    den2=0;
    for f=px-ancho:1:px+ancho
        for g=py-alto:1:py+alto
            SSD=SSD+(imagen_left(f,g)-imagen_right(f,g-h))^2;
            den2=den2+(imagen_right(f,g-h))^2;
        end
    end
    den2=sqrt(den2);
    cr1=SSD/(den1*den2);
    Cr_vect(h)=cr1;
end

```


- **Cálculo de la correlación por el método ZNSSD.**

```

function [Cr_vect]=znssd(ancho,px,py,imagen_left,imagen_right,barrido)
% Cálculo de la correlación mediante el método de la suma normalizada
de diferencias cuadradas con media cero, ZNSSD
% No usado como solución final para el cálculo de la correlación.

cr1=0; Cr_vect=zeros(1,barrido,'double');
media_l=0; den1=0; alto=ancho;

%Cálculo de la intensidad media de la imagen izquierda sobre la
ventana de vecindad.
for f=px-ancho:1:px+ancho
    for g=py-alto:1:py+alto
        media_l=media_l+imagen_left(f,g);
    end
end
media_l=media_l/((2*ancho+1)*(2*alto+1));
for f=px-ancho:1:px+ancho
    for g=py-alto:1:py+alto
        den1=den1+(imagen_left(f,g)^2-media_l)^2;
    end
end
den1=sqrt(den1);

for h=1:barrido
    den2=0;SSD=0;media_r=0;
    %Cálculo de la intensidad media de la imagen derecha sobre la
ventana de vecindad
    for f=px-ancho:1:px+ancho
        for g=py-alto:1:py+alto
            media_r=media_r+imagen_right(f,g-h);
        end
    end
    media_r=media_r/((2*ancho+1)*(2*alto+1));

    for f=px-ancho:1:px+ancho
        for g=py-alto:1:py+alto
            SSD=SSD+((imagen_left(f,g)-media_l)-(imagen_right(f,g-h)-
media_r))^2;
            den2=den2+(imagen_right(f,g-h)^2-media_r)^2;
        end
    end
    den2=sqrt(den2);
    cr1=SSD/(den1*den2);
    Cr_vect(h)=cr1;
end
end

```

ANEXO 2. HOJAS DE CARÁCTERÍSTICAS

A continuación se muestran las hojas de características de los dispositivos empleados en el sistema de adquisición de imágenes.

1) TECHNICAL CHARACTERISTICS



The ARvision-3D is a specialized product designed for Augmented Reality applications. It is an opto-electronic device that projects an image or a streaming video through the near-the-eye micro-displays.

a) Display Features

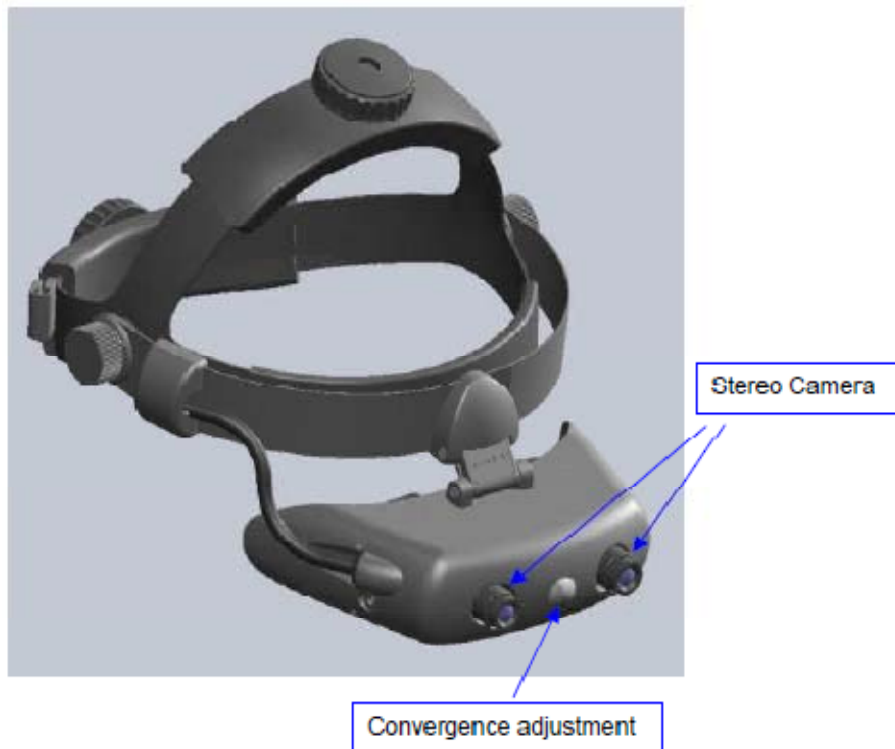
The ARvision-3D offers micro-screens with the following characteristics:

- 800 x 600 resolution, equals 1.4M pixels and full color.
- Displays are Liquid Crystal (LCD), one of the best quality of image. The image has an excellent sharpness, brightness, and contrast which allows a comfortable reading visibility without getting your eyes tired.
- Brightness and contrast are adjusted by computer, through a software, provided for the device.

b) Camera Features

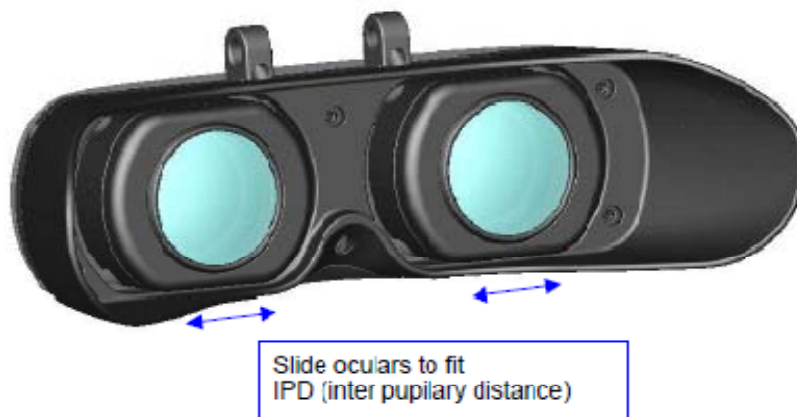
The cameras we offer for the Augmented Reality range of products are:

- USB 2.0
- IEEE 1394 (FireWire)
- 752 x 480 resolution, 61fps, color or b/w
- Progressive scan 1/3" CMOS, global shutter
- Camera control by software
- Gain control: automatic / manual, 0 dB to 12 dB
- Shutter Speed: automatic / manual, 0.12 ms to 512 ms
- Synchronization: external trigger, software trigger, or free-running
- Power requirement: 5V (USB), <1 Watt
- Standard lens 6mm, changeable by user, lens mount M12
- Convergence adjustment



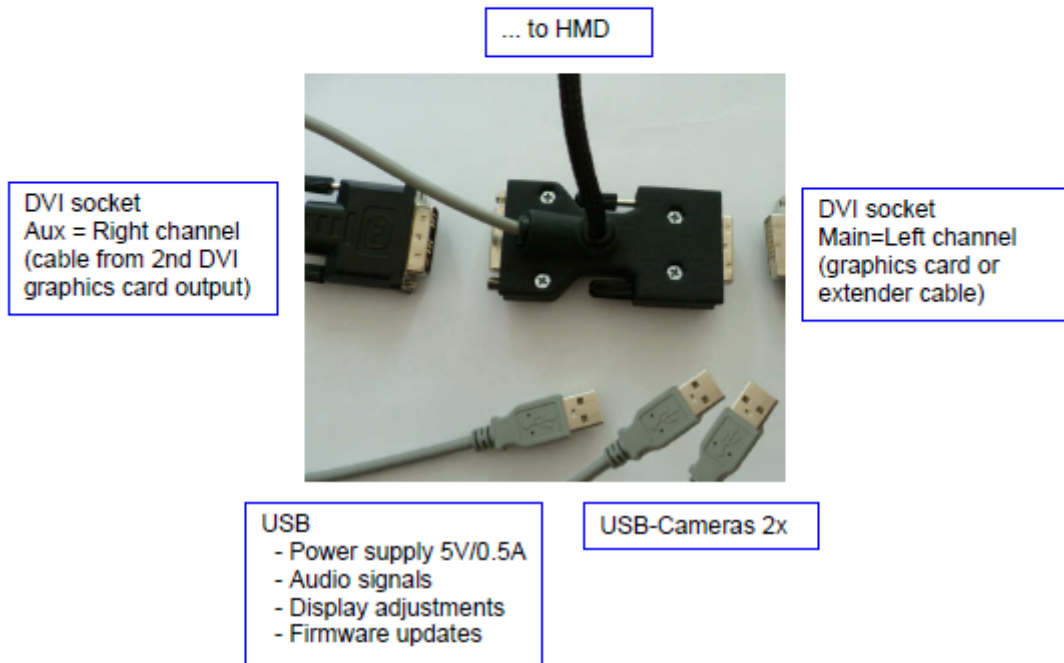
c) Attachment and Mechanical Adjustment

- Head Band: The ARvision-3D-HMD



d) Connectivity and GPU Support

Any computer, laptop or other video source with digital DVI output is suitable



e) Stereoscopic Signals

The ARvision-3D-HMD offers two stereo options:

- Passive stereo (two DVI channels)
- Active Stereo

f) Audio

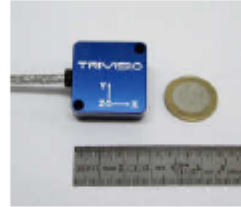
- Microphone is integrated
- Earphone jack (3.5 mm) for external earphones
- Audio signals by USB

II) Additional Features and Options

a) Inertial Motion Sensor

Upon request we can integrate inertial motion sensors:

- Wireless Colibri
- Colibri (with USB cable)



b) Cable Length

The HMD comes with a basic cable length of 2.2 meters, but can be extended upon request to 5m.

c) VGA Interface

There will be a variant with analog VGA video interface, final development stage Dec 2010.

III) Overview

Technical Specifications	Cameras 2x	752x480 (max 61fps), USB 2.0 (optional IEEE1394)
	Camera Sensor	1/3" CMOS, progressive scan, global shutter
	Gain	auto/manual, 0 dB to 12 dB
	Shutter Speed	auto/manual, 0.12 ms to 512 ms
	Synchronization	external trigger, software trigger, or free-running
	Convergence	manually adjustable
	Focus	manually adjustable
	Iris	fixed
	Standard Lens	6mm (0.9x vision), changeable by user, lens mount M12
	Microdisplays 2x	SVGA AMLCD 800x600 color (equals 1,440,000 pixels)
	Display Color	24 bit
	Frame Rate	60fps
	Overlap	100%
	Field of View	42° diagonal (4:3, 34° (horiz), 25° (vert))
	Distortion	3.1% (horiz), 1.9% (vert), 4.0% corner
	Eye distance (IPD)	55-72mm adjustable
	Eye Relief	27mm
	Eye Motion Box	7mm (horiz) x 6mm (vert)
	Video Interface	DVI-D
	Stereoscopic Modes	Mono, passive stereo, active stereo
	Audio	Microphone integrated, 3.5mm socket for headset (via USB)
	Cable Length	2.2m (optional 5m)
	Power Consumption	Display 2.5W, Camera <1W each (5V, taken from USB port)
Operating Temperature	0°C to +40°C	
Weight	290 g (without headband)	
Dimensions (W/H/D):	150mm x 42mm x 84mm	

Subject to technical modifications

TRIVISIO

User Manual

ARvision-3D-DVI

(with USB 2.0 or IEEE1394 FireWire Cameras)



Subject to technical modifications

Copyright: Trivisio Prototyping GmbH, Karcherstrasse 10, D-67655 Kaiserslautern

WARNING

Before using the ARvision-3D HMD please read carefully these safety instructions.

Read this handbook and use the ARvision-3D device in strict accordance in order to prevent any damage to your eye, other injury, loss of visual functions, property damage or death.

Children under the age of fifteen may not use this product.

We strongly advise you to get familiar with the ARvision-3D device and its capabilities before you use it for the first time.

Table of Contents

Table of Contents..... 3

1 Safety Instructions 4

1.1 Health Concerns.....4

1.2 General Safety Hints5

1.3 Warning for Electrical Products.....5

2 Description 6

2.1 Head Mounted Display HMD6

2.2 DVI Dongle.....7

3 General Operating Instructions 8

3.1 Installing Software.....8

3.2 Connecting External Video Sources.....8

3.3 Attachment and Mechanical Adjustment9

3.4 Adjusting the Interpupil Distance10

3.5 Adjusting the Image (by Software).....11

3.6 USB-Camera / Software12

3.7 AR Setup.....12

3.8 Using Different Stereo Modes.....14

3.9 Using Audio.....14

3.10 Adjusting Cameras.....15

3.11 Firmware update.....16

4 Maintenance and Cleaning Instructions..... 17

5 Troubleshooting 18

6 Returning used electronic devices in EU countries 19

7 Technical Data 20

1 Safety Instructions

1.1 Health Concerns

→ **WARNING: Eye Disease, Eye Injury, and Glaucoma**

If you have been diagnosed with or are susceptible to eye disease, eye injury, or glaucoma consult your doctor before use and do not use without your doctor's approval.

→ **CAUTION: Heart Disease, High Blood Pressure**

If you have a history of heart disease or high blood pressure consult your doctor before use and do not use without his approval.

→ **CAUTION: Seizures**

If you have a history of temporary spasm, unconsciousness, or epileptic seizures from light stimulation, consult your doctor before use and do not use without his approval.

If any of the following symptoms occur, if using ARvision-3D, stop using immediately and rest:

- eye fatigue or irritation,
- headaches or dizziness,
- aches and pain in the neck or shoulders,
- double vision,
- nausea or motion sickness,
- inability to focus on the displays.

Misuse or overuse of this product may result in eye damage, or loss of visual functions.

1.2 General Safety Hints

- Avoid getting the cable entangled around your neck, body or arms. Use the belt pouch for the power unit and wear the cable close to your body.
- To ensure that the device does not fall off during use, always use the headband, and have it tightened with an adequate tension.

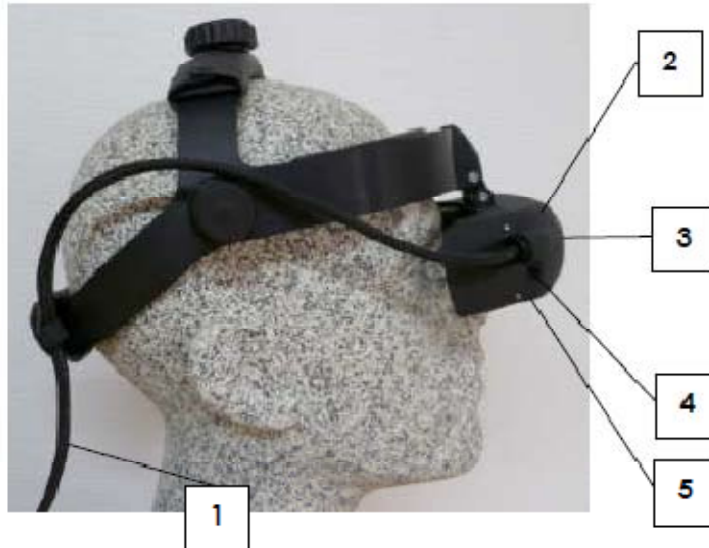
1.3 Warning for Electrical Products

- To avoid any risk of electrocution, do not bring any part of the ARvision-3D in contact with water when the device is connected to an AC outlet (e.g. when plugged to PC).
- Avoid using and storing the ARvision-3D at wet, humid, dusty and smoky surroundings and extreme temperatures.
- Do not use ARvision-3D at temperatures below 0°C and above +40°C. Avoid dropping or mechanical shock, as frame and displays may be deformed.
- Always switch off and unplug the ARvision-3D when it will not be used.
- In case of damage contact your retailer. There are no user serviceable parts. Only qualified service personnel should perform any service required on this product.

2 Description

2.1 Head Mounted Display HMD

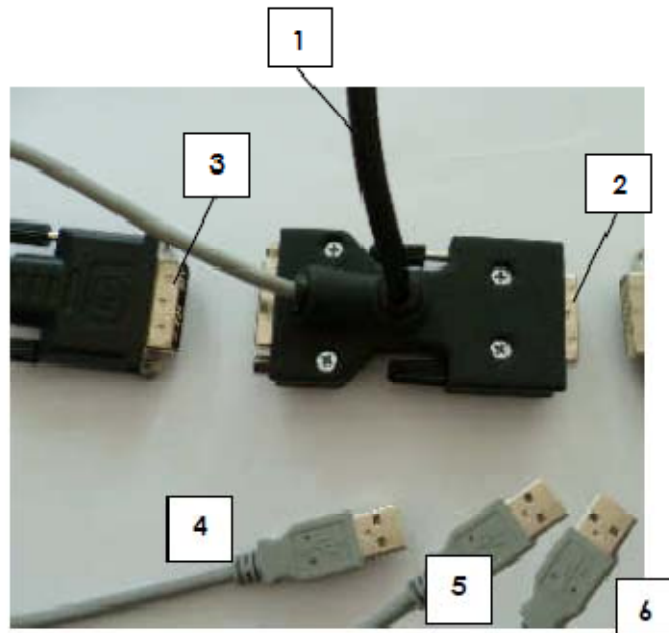
The HMD contains two **microdisplays** (2) to show the images. The **cable** (1) is permanently attached to the HMD and to the dongle element. The built-in color cameras (3) provides the video signal to the power unit for output to a PC. Without a PC or external video source connected to the device there is no image visible inside the HMD. Into the 3.5mm socket (4) earphones can be connected. There is a built-in microphone (5) on the bottom side of the case.



2.2 DVI Dongle

The DVI dongle contains the following elements:

- (1) HMD cable permanently attached
- (2) DVI connector, Main=Left channel
(connect to graphics card)
- (3) DVI socket, Aux=Right channel
(connect by cable from 2nd DVI graphics card output)
- (4) USB cable for power supply, audio and settings
- (5) USB cable left camera
- (6) USB cable right camera



3 General Operating Instructions

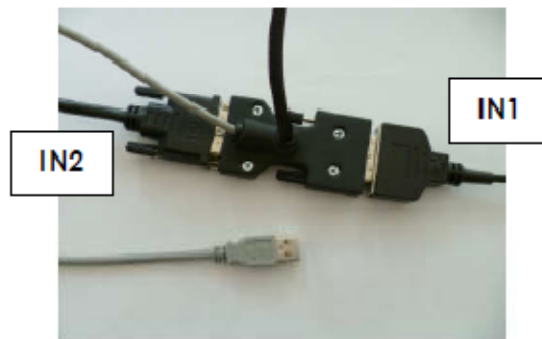
3.1 Installing Software

Please install the camera drivers and HMD adjusting software before you connect the device the first to time to your PC.

3.2 Connecting External Video Sources

To prevent damage to the electronic, first connect the video cables with your PC before you power on the device by connecting the USB cable. You can connect the device by plugging the DVI dongle directly into your graphics card (in case of HDMI output you can use converter plug) or by using DVI extender cable. This gives you Input1 (IN1) what is the main channel. You can also connect a second DVI signal to the Aux channel (IN2) from second DVI output of the same graphics card or from another graphics card, see also stereoscopic options "3.8 Using Different Stereo Modes" page 14). The device will automatically recognize the correct signal.

A static image can burn into the microdisplays after an extended period of time.



3.3 Attachment and Mechanical Adjustment

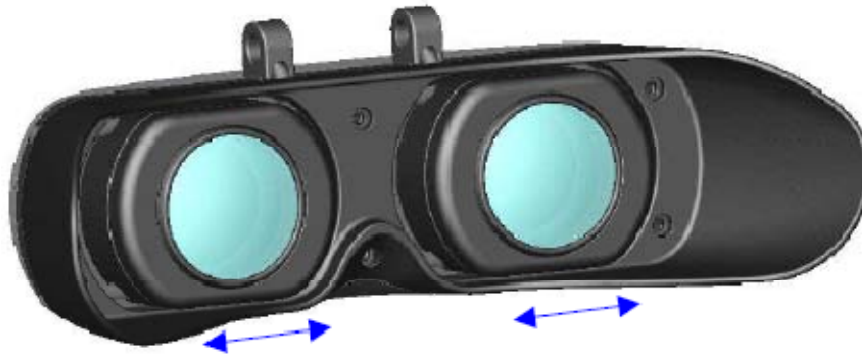
Putting on the HMD

- Put on the ARvision HMD. Place the headband around the back of your head and pull it just tight enough to make it fit safely and comfortably.
- On the displays inside the HMD, you can now see the pictures of your video sources.
- Wearing the HMD you will see your surroundings only constricted. For this reason, please, move only carefully or not at all.



3.4 Adjusting the Interpupil Distance

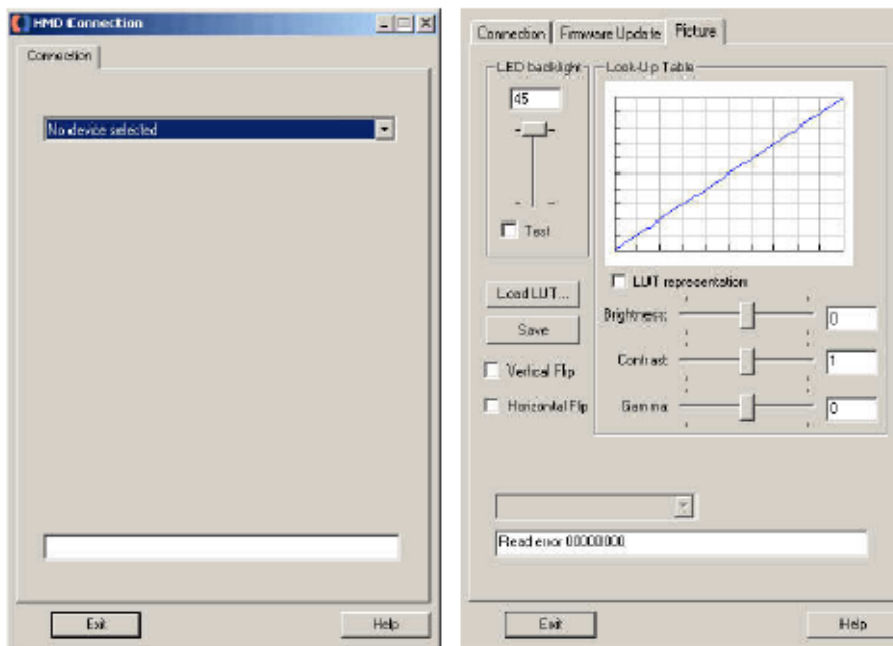
To optimize the position of the two displays, so that both images are seen as a single visual field, carefully move the oculars into the right position.



Slide oculars to fit
IPD (inter pupillary distance)

3.5 Adjusting the Image (by Software)

To adjust the image start the software "hmd_connection.exe" which can be downloaded from www.trivisio.com. Please install **first** the software according instructions **before** you connect the HMD with the PC using the DVI and USB-cable. After installation process plug in the DVI and USB cable to the PC to turn on the device. Windows will detect the hardware and install automatically. To adjust the image start the software "hmd_connection.exe" either from icon or start menu. Then select the HMD (also serial number is shown here). Then you can go to menu "Picture" and adjust the brightness, contrast and gamma of the image. You can also change the look-up table (LUT) or flip the image.

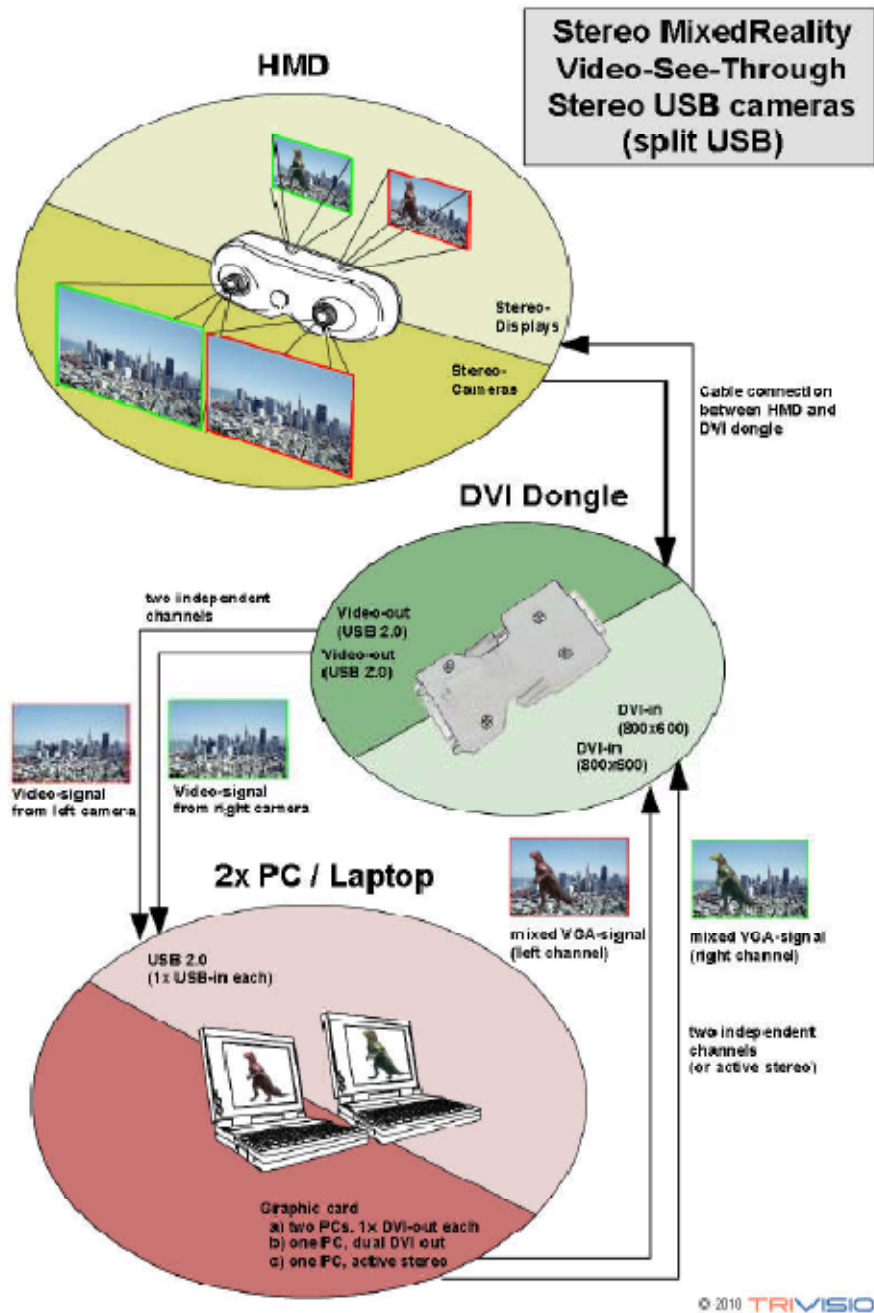


3.6 USB-Camera / Software

Please install **first** the camera driver software downloaded from www.trivisio.com according instructions **before** you connect the power unit with the PC using the USB-cables. Then power on the device and Microsoft Windows® will recognize new hardware automatic. To check both USB cameras you can run the CameraViewer software twice. **To avoid damage (and to prevent aging) please unplug the USB-cables between HMD and PC during lightning storm and when unused for long periods of time.**

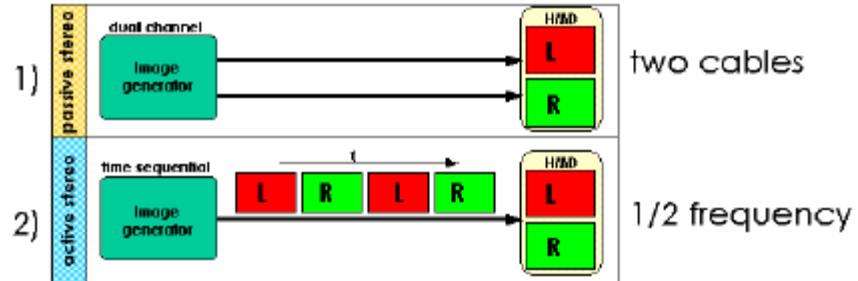
3.7 AR Setup

To use the device with AR function (augmented reality) the camera signal must be sent to a computer. Then the overlaid stereoscopic video signal from the PC can be directed into the power unit using one or two DVI cables (2). The PC-resolution must be set to 800x600@60Hz.



3.8 Using Different Stereo Modes

The electronics of your ARvision-3D device allows two different stereo modes. The electronic is automatically detecting when active stereo signal is supplied by the graphics card.

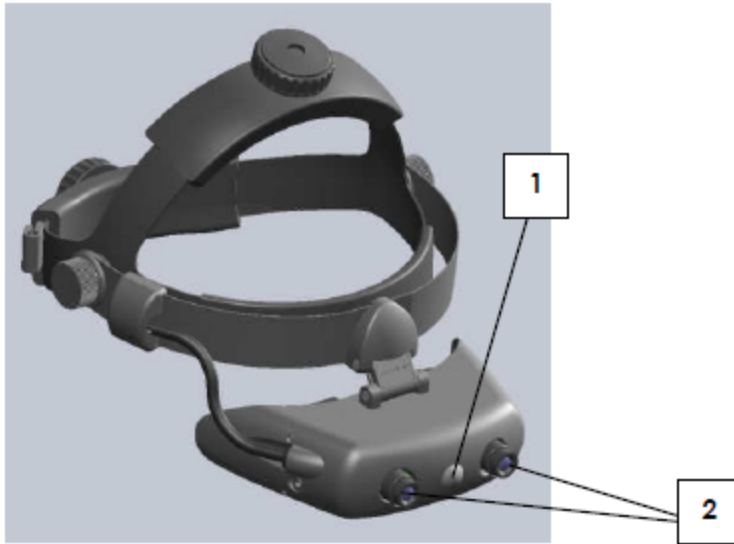


3.9 Using Audio

After connecting the USB cable to your PC, Windows will detect the built-in microphone and also the earphones as audio devices.

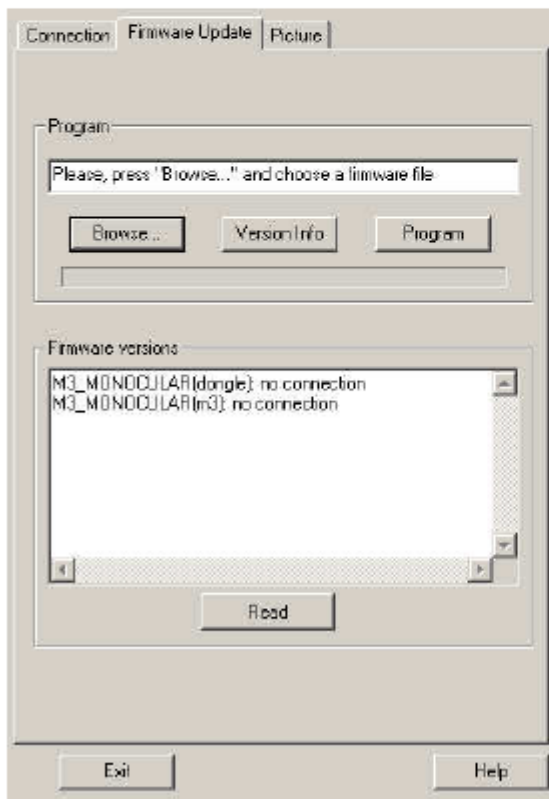
3.10 Adjusting Cameras

For adjusting the convergence use the wheel (1). The focus can be manually adjusted by turning the camera lenses (2).



3.11 Firmware update

In the need of a firmware update connect the device by a USB cable to the PC and turn on the device. Then run HMD-software "hmd_connection.exe" and choose the menu of new firmware. Select the firmware file by clicking on "Browse". The firmware is provided as "RIGHT_0x000?.hex" and "LEFT_VGA_DVI_0x000?.hex" files. Then click "Program". You should program these files one by one in above mentioned order. After programming the last file the device will turn off automatically. **During the update process never unplug the device and do not shut down the computer and do not turn off the power!** If the device fails to turn on, try to reprogram "LEFT_VGA_DVI_0x000?.hex" file, holding power-button pushed during programming process.



4 Maintenance and Cleaning Instructions

To clean the HMD and control unit, use a damp cloth. A light rinsing agent may be added to the water. Do not use any detergents.

To clean the oculars of the HMD use a dry, clean cloth.

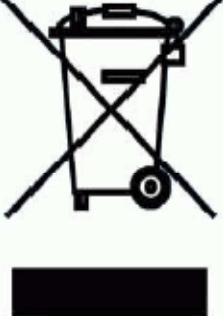
In case of damage contact your retailer. There are no user serviceable parts. Only qualified service personnel should perform any service required on this product.

5 Troubleshooting

Problem	Cause	Remedy
Only dark, or only white image	Incorrect brightness.	See HMD software and adjust correct
No image	No DVI signal applied	Connect a video source (SVGA)
No image	Wrong resolution or frequency from graphics card	Set 800x600@60Hz
No image	Appliance switched off, no power connected	Plug USB cable to PC
Unsharp image	Focus of camera lens is not adjusted	Adjust the focus by turning the camera lens

All other problems should be performed by a trained and authorised service provider !

6 Returning used electronic devices in EU countries

	<p>This HMD and accessories shall not be treated as household waste. The separate collection is a condition for reuse, recycling and utilisation of used electronic devices, which ensures the protection of resources. To comply with german ElektroG (Rücknahme und Entsorgung von Elektro- und Elektronikaltgeräten) and european WEEE (Waste Electrical and Electronic Equipment) electronic used devices from private households can be returned free of charge. For details please contact your local distributor or directly Trivisio Prototyping GmbH (www.trivisio.com).</p>
---	---

7 Technical Data

Cameras 2x	752x480 (max 61fps), USB 2.0 (optional IEEE1394)
Sensor	1/3" CMOS, progressive scan, global shutter
Gain	auto/manual, 0 dB to 12 dB
Shutter Speed	auto/manual, 0.12 ms to 512 ms
Synchronization	external trigger, software trigger, or free-running
Convergence	manually adjustable
Focus	manually adjustable
Iris	fixed
Lens	6mm (0.9x vision), changeable by user, lens mount M12
Microdisplays 2x	SVGA AMLCD 800x600 color (equals 1,440,000 pixels)
Display Color	24 bit
Frame Rate	60fps
Overlap	100%
Field of View	41.8° diagonal (4:3, 33.7° (horiz), 25.2° (vert))
Distortion	3.1% (horiz), 1.9% (vert), 4.0% corner
Eye distance (IPD)	55-72mm adjustable
Eye Relief	27mm
Eye Motion Box	7mm (horiz) x 6mm (vert)
Video Interface	DVI-D
Stereoscopic Modes	Mono, passive stereo, active stereo
Audio (via USB)	Microphone integrated, 3.5mm socket for headset
Cable Length	2.2m (optional 5m)
Power Consumption	Display 2.5W, Camera <1W each (5V, taken from USB port)
Operating Temperature	0°C to +40°C
Weight	290 g (without headband)
Dimensions (W/H/D):	150mm x 42mm x 84mm

Subject to technical modifications



Guía de Usuario de Webcámara Minoru 3D

Introducción

Conozca a Minoru, lo cual significa "Realidad" en Japonés, la primera webcámara mundial en 3D.

Conecte la Minoru a su PC y ésta comenzará a vivir, se le iluminarán los ojos y comenzará a mirarle en 3D.

Sus amigos y familia comenzarán a verle ahora en 3D en su programa favorito de mensajería, tal como el Windows Live Messenger, Skype, AOL instant messenger, OoVoo y muchos otros.

Podrá tomar fotos en 3D o incluso grabar videos en 3D y podrá descargarlos a YouTube.

Instalación

Instalación de Hardware

Desempaquetar la Minoru cuidadosamente y ponerla sobre la mesa de despacho delante de usted o ponerla encima de su monitor como se muestra abajo. Idealmente la Minoru deberá colocarse aproximadamente a 3 pies (1 metro) de distancia de usted.



Conecte la Minoru en un portal USB 2.0 de alta velocidad de su PC. Si es posible, por favor, use uno de los portales USB detrás del PC.

Nota: ¡La Minoru no funcionará adecuadamente si se conecta a un antiguo tipo de portal USB 1.1!

Una ventana resaltante deberá aparecer encima de la bandeja del sistema diciendo, 'Se Encontró Nuevo Hardware' o 'Instalando el Software de Accionamiento de Dispositivo'. Espere a que se complete este proceso antes de instalar el software Minoru.

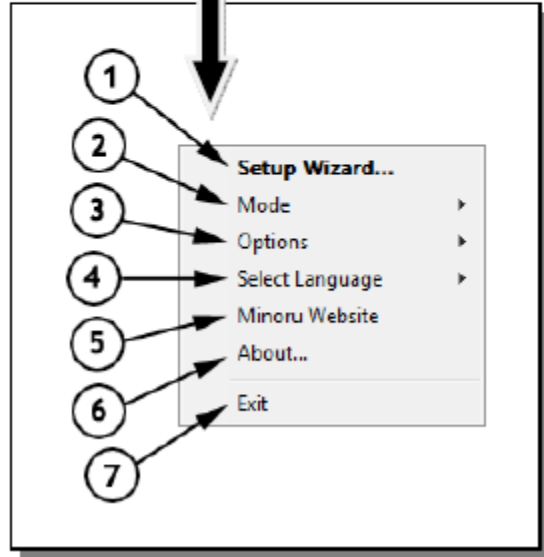
Instalación de Software

Abra el portador de CD ROM y ponga el CD de software en el portador. Cierre el portador de CD ROM y siga las instrucciones en pantalla para completar la instalación del software Minoru software.

Configuración de la Minoru

Podrá acceder las configuraciones de la Minoru en cualquier momento haciendo un clic de derecha en el icono de estado de la Minoru en la bandeja de sistema. Usted podrá acceder y cambiar las configuraciones incluso cuando la Minoru esté en el medio de una video llamada o durante una sesión de video grabación. Un doble clic en el icono inicia el programa de ayuda de configuración.

Si no aparece el icono de estado de la Minoru en la bandeja de sistema, deberá iniciar la 'Configuración de Webcámara Minoru' haciendo un doble clic en el atajo en pantalla.



1 - Programa de Ayuda de Configuración...

Antes de usar la Minoru por primera vez deberá enfocar y alinear los dos ojos de la Minoru



Ponga primero la Minoru sobre la mesa de su despacho o el monitor a una distancia equivalente a 3 pies (1 metro) de su cara. En esta fase no deberá tener puestas las gafas 3D.



Enfoque el ojo izquierdo de la Minoru girando el aro del lente hasta que haya una buena y clara imagen de su cara.



Enfoque el ojo derecho de la Minoru girando el aro del lente hasta que haya una buena y clara imagen de su cara.



Alineación vertical... En esta fase todavía no deberá tener puestas las gafas 3D. Mire la Minoru, deberá ver dos imágenes de su cara en la pantalla, una imagen azul y otra roja. Ajuste el deslizador vertical de manera que las imágenes azul y roja de su cara estén exactamente a la misma altura en la pantalla. A veces, es más fácil mirar a la parte superior o inferior de su cabeza o a sus ojos para alinear las dos imágenes.



Alineación horizontal... Todavía no se ponga las gafas 3D, no se preocupe, las podrá poner pronto. Mire a las dos imágenes de su cara en la pantalla y ajuste el deslizador horizontal de manera que las dos imágenes de su cara se junten y formen una imagen. Tenga en cuenta que todavía verá imágenes azules y rojas de las cosas delante y detrás de usted.



Ahora ponga las gafas 3D por primera vez. Pase algún tiempo experimentando el efecto 3D por sí mismo. Estire sus brazos hacia delante y parecerá que salen de la pantalla. Ponga sus brazos detrás de su cabeza y parecerá que entran en la pantalla

2 - Modo

Puede seleccionar de tres modos de operación distintos para la Minoru.

3D - El visualizador de vídeo 3D necesitará usar las gafas 3D para poder ver el efecto 3D.

Imagen en Imagen (PiP) - Este modo es valioso si desea mostrar detalles cercanos de algo mientras mantiene la visión de su cara. Coloque la cosa en el ángulo de visión del ojo izquierdo y se mostrará una imagen detallada en la ventana grande PiP.

2D - Puede usar este modo si el visualizador no lleva gafas 3D.

Lado a Lado (SBS) ¡Para un entusiasmo real de 3D! Seleccione de Izquierda/Derecha (LR) o Derecha/Izquierda (RL). Podrá utilizar estos modos para grabar imágenes 'Lado a Lado'. El vídeo grabado podrá entonces retransmitirse usando un retransmisor estereoscópico, véase 'Software recomendado para obtener lo mejor de la Minoru', página ??

El modo actual se indica en el icono de estado de la Minoru en la bandeja de sistema.



- Minoru trabajando en modo 3D



- Minoru trabajando en modo Imagen en Imagen (PiP)



- Minoru trabajando en modo 2D



- Minoru trabajando en modo 'Lado a Lado' (SBS)



- Minoru sin conectar al PC. Por favor comprobar que el conector USB de la Minoru está conectado en el PC.

3 - Opciones

Iniciar Minoru al iniciarse Windows. Si se marca esto, entonces la Configuración de Webcámara Minoru se inicia automáticamente al iniciarse Windows.

4 - Selección de Idioma

Seleccione el idioma que va a usarse para el programa de ayuda de configuración de la Minoru y la aplicación de grabación.

5 - Páginas Web de la Minoru

Haga clic para ir directamente a las Páginas Web de la Minoru 3D.

6 - Información

Se muestra la información de la versión de software Minoru. Por favor, tome nota del número de la versión para indicarlo en cualquier comunicación con el apoyo de cliente de la Minoru.

7 - Salida

Haga clic para salir de la 'Configuración de Webcámara Minoru'. La Configuración de Webcámara podrá reiniciarse haciendo un doble clic en el icono de atajo de 'Configuración de Webcámara Minoru' en la pantalla.

Aplicación de grabación



1 - Reinicio de la Minoru

Cuando inicie la 'Captura de Vídeo Minoru' puede que vea el mensaje de información indicado abajo.



La Minoru está siendo usada probablemente por otra aplicación, tal como el software de mensajería, o por el 'Programa de Ayuda de Configuración de Minoru' en funcionamiento. Cierre la otra aplicación que está usando la Minoru o cuelgue la llamada de video mensaje, entonces haga clic en el botón de 'Reinicio de la Minoru'. La toma de la Minoru deberá aparecer entonces en la pantalla.

2 - Resolución

Seleccionar la resolución de la Minoru. Elegir de 320 x 240, 352 x 288, 640 x 480 ó 800 x 600 pixeles. La resolución elegida se usa para el vídeo grabado o la foto tomada.

3 - Modo de Pantalla

Seleccionar el modo de pantalla para el pre visualizador de vídeo. Elegir de 100%, 200% o Pantalla completa. Cuando mire vídeo 3D en pantalla completa, es preferido estar a distancia del monitor, 6 pies (2 metros) es la distancia recomendada para ver.

4 - Selección de entrada audio

'Novo - Minoru's Ears' se usa por configuración automática para la grabación de la banda sonora del archivo de vídeo. Usted puede seleccionar un dispositivo distinto de grabación, si prefiere usarlo, en vez de la configuración automática.

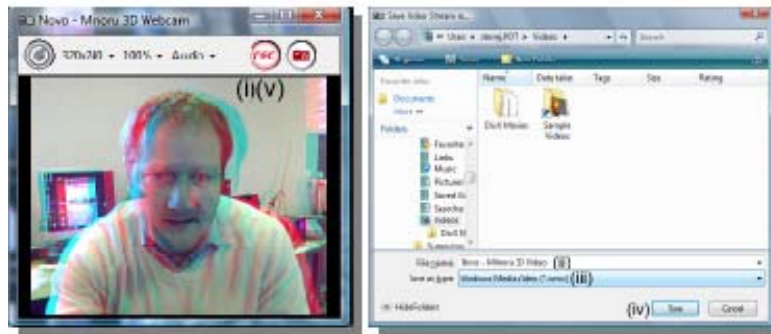
5 - Inicio/Paro de Grabación

Este botón se usa para grabar el vídeo; vea lo indicado abajo para aprender cómo grabar un vídeo.

6 - Toma de Foto

Este botón se usa para tomar una foto, vea las indicaciones abajo para aprender cómo tomar una foto y enviarla por email.

Cómo grabar y playback un vídeo 3D



Primero deberá seleccionar la resolución del vídeo. Después, deberá componer la imagen y hacer clic en el botón de 'Comenzar a Grabar' (i). Entonces se le pedirá que elija un nombre de archivo (ii) y que también elija el tipo de archivo (iii).

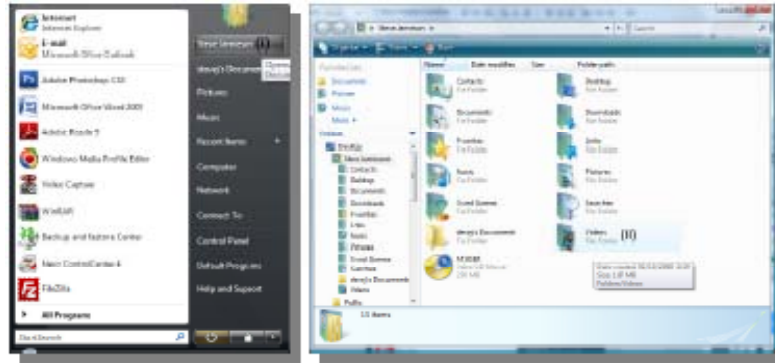
Los dos tipos de archivo disponibles son .asf y .avi

Elija .asf si desea un menor tamaño de archivo apropiado para poder descargar a YouTube y otras páginas web de redes sociales. El archivo tipo .asf se selecciona como configuración automática.

Elija .avi para grabar en la mejor calidad 3D. Estos archivos vídeo sin comprimir serán muy grandes y no serán adecuados para descargar. Estos archivos podrían agotar pronto su disco duro, por tanto, es preferible usar el tipo .asf en vez de éste, lo más posible.

Clic en 'Salvar' (iv) y comenzará la grabación vídeo. Para parar la grabación, haga clic en el botón de 'Parar la Grabación' (v).

Por configuración automática, los videos serán grabados en el directorio 'Vídeos' o 'Mis Vídeos' del disco duro.



Para playback del vídeo grabado, deberá hacer uno de lo siguiente:

En el sistema Vista, haga clic en Inicio..(Su nombre de usuario) (i), y entonces haga un doble clic en Vídeo (ii). Elija uno de los vídeos y haga un doble clic en él para playback.

En el sistema XP, haga clic en Inicio..Mis Documentos (i), y entonces haga un doble clic en Mis Vídeos (ii). Elija uno de los vídeos y haga un doble clic en él para playback.

Cómo tomar y enviar una foto por email

Primero deberá seleccionar la resolución de la foto. Después, deberá componer la imagen y hacer clic en el botón de 'Tomar una Foto'. La foto aparecerá irmediatamente en una ventana independiente y entonces usted tendrá la elección de (i) salvar o (ii) enviar la imagen por email.



Si decide salvar la imagen, entonces haga clic en el botón de 'Salvar Foto' (i). Podrá elegir salvar el; archivo en bitmap (.bmp), jpeg (.jpg), imagen marcada (.tiff) o gráficos de red portátil (.png). Para obtener la mejor calidad 3D deberá seleccionar .bmp o .tiff. También podrá elegir donde va a guardar la imagen antes de hacer clic en 'Salvar'.

Si elige enviar la imagen por email, entonces haga clic en el botón de 'E-mail la Foto' (ii). Se abrirá inmediatamente un nuevo email con la imagen adjunta en bitmap (.bmp) ya preparada para enviar. Solamente tiene que teclear la dirección de email del recipiente y hacer clic en enviar. La imagen será guardada automáticamente en su PC en la última selección de salvar.

Investigación de Averías

Problema

- La imagen de la webcámara se congela o la webcámara no funciona en los modos de alta resolución 640 x 480 y 800 x 600.
- Cuando se conecta la Minoru en un portal USB delantero se ve un mensaje, 'No se conoce el dispositivo USB'.

Solución

- Conectar la Minoru en un portal USB trasero del PC. Los portales USB delanteros de algunos PCs no funcionan adecuadamente a la alta especificación requerida por la Minoru.
- Algunos PCs antiguos más lentos no pueden mostrar el vídeo lo suficientemente rápido cuando se usan los modos de alta resolución. Por favor use el modo 320 x 240 en vez de los otros.

Problema

- La Minoru no funciona con Yahoo Messenger

Solución

- Yahoo Messenger usa una antigua tecnología vídeo y la Minoru no es compatible con esto.

Para más ayuda por favor visite www.minoru3d.com

Especificaciones

USB 2.0 de Alta Velocidad para Webcámara 3D

2 x VGA 640x430 CMOS sensor

2 x lentes de ángulo ancho de alta calidad

Micrófono USB incorporado

Imagen de 800 x 600, 640 x 480, 352 x 288 y 320 x 240 píxeles, hasta 30 fps

Imagen - Anaglifo 3D optimizado (rojo/cian), Imagen en Imagen (PIP) y 2D estándar

Pedestal Multi posición

Compatible con Microsoft Windows XP y Microsoft Windows Vista

Declaración de Cumplimiento CE



Este producto ha estado sujeto a los requerimientos de pruebas necesarios para obtener la marca CE, y ha sido aprobado para poder llevar la marca del estándar CE,

EN 55022:2006 + A1:2007

EN 55024:1996 + A1:2001, + A2:2003

La vigencia de la marca CE está restringida a aquellos países donde puede ser legalmente impuesta.

Declaración FCC



Este dispositivo cumple la Parte 15 de las reglas FCC. Su operación está sujeta a las dos condiciones siguientes: (1) que este dispositivo no cause interferencias ofensivas, y (2) que este dispositivo deba poder aceptar cualquier interferencia recibida, incluyendo interferencias que puedan causar una operación indeseada.

Instrucciones de desecho de producto (Unión Europea)

El símbolo mostrado aquí y en el producto significa que el producto está clasificado como Equipo Eléctrico o Electrónico y no deberá tirarse con la basura de casa o basura comercial al final de su vida útil.

La Directiva de Desechos de Equipo Eléctrico o Electrónico (WEEE) (2002/96/CE) ha sido introducida para reciclar los productos usando las mejores técnicas de recuperación y reciclaje y minimizar el impacto medio ambiental, tratar cualquier sustancia peligrosa y evitar aumentar el relleno de terrenos.



Cuando no tenga más uso para este producto, por favor deséchelo usando los procesos de reciclaje indicados por su autoridad local. Para más información, por favor contacte a su autoridad local o al comercio minorista donde adquirió este producto.

Todas las marcas registradas son propiedad de sus dueños respectivos.

GLOSARIO

- AR *Augmented reality (Realidad Aumentada).*
- CRT *Cathode Raid Tube.*
- DVI *Digital Visual Interface.*
- FFT *Fast Fourier Transformation.*
- GUIDE *Graphical User Interface Development Enviroment.*
- HDMI *High Definition Multimedia Interface.*
- HOE *Holographic Optical Element.*
- HMD *Head-Mounted Display.*
- IR *Radiación infrarroja (Infrared).*
- LCD *Liquid Crystal Display.*
- LCoS *Liquid Crystal on Silicon.*
- LED *Light-Emitting Diode.*
- NCC *Normalized Cross-Correlation.*
- NSSD *Normalized Sum of Squared Differences.*
- OLED *Organic Light-Emitting Diode.*
- SAD *Sum of Absolute Differences.*
- SSD *Sum of Squared Differences.*
- VR *Virtual Reality (Realidad Virtual).*
- ZNCC *Zero-mean Normalized Cross-Correlation.*

ZSAD *Zero-mean Sum of Absolute Differences.*

ZSSD *Zero-mean Sum of Squared Differences.*

ZNSSD *Zero-mean Normalized Sum of Squared Differences.*