



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 5

CENTRO DE INFORMÁTICA INDUSTRIAL

**POSICIONES RELATIVAS DE PERSONAS EN UNA ESCENA A PARTIR DEL  
RECONOCIMIENTO FACIAL.**

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

**Autor(es):**

Lislien González Rodríguez

Andiel Muñoz Viamonte

**Tutor:** MSc. Yuniesky Coca Bergolla

**Co-tutor:** Ing. Adrián Fonseca Mursulí

La Habana, 2013

## Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Lislien González Rodríguez

\_\_\_\_\_  
Firma del Autor(a)

Andiel Muñoz Viamonte

\_\_\_\_\_  
Firma del Autor(a)

MSc. Yuniesky Coca Bergolla

\_\_\_\_\_  
Firma del Tutor

Ing. Adrián Fonseca Mursulí

\_\_\_\_\_  
Firma del Co-tutor

## Datos de Contacto

**Tutor:** MSc. Yuniesky Coca Bergolla

**Empresa:** Universidad de Ciencias Informáticas UCI.

**Dirección:** Carretera a San Antonio Km. 2 1/2 Torrens, Boyeros La Habana.

**E-mail:** [ycoca@uci.cu](mailto:ycoca@uci.cu)

**Tutor:** Ing. Adrián Fonseca Mursulí.

**Empresa:** Universidad de Ciencias Informáticas UCI.

**Dirección:** Carretera a San Antonio Km. 2 1/2 Torrens, Boyeros La Habana.

**E-mail:** [amursuli@uci.cu](mailto:amursuli@uci.cu)

## Agradecimientos

### **Lislien**

*Luego de tantos años de dedicación y estudio, me despidió de mi vida como universitaria, no sin antes agradecerle a todos aquellos que de una forma u otra estuvieron presentes y dieron su apoyo:*

***A mi familia...**en especial a mi abuelita Luisa, a mi mamá, a mi tío Félix, a mis tías Marle, Martha, Niurka y Greylis, a mi padrastro Jesús, a mi hermana Sianis, a mis primos, Eduardo, Félix Jorge, Lisdán, Carlos E., Mayvis, Carlitín y Alejandro; por darme todo su apoyo incondicional y mucho más.*

***A mi novio Yaxniel,** por ser mi amigo, mi compañero y ser la personita que ha estado siempre a mi lado, dándome todo el apoyo del mundo en estos 5 años de carrera, gracias por dejarme entrar en tu corazón.*

***A mi amiga Yuned,** mi "ranita blanca", por ser la que me apoyó y convenció para que eligiera esta carrera, gracias por estar siempre y poder contar con tu apoyo cuando lo necesito, te quiero un montón.*

***A Leonel,** gracias por tu amistad, por apoyarme aun estando lejos, por ayudarme a ver soluciones cuando yo creía que no las había y sobre todo por esos regaños cada vez que me los merecía.*

***A mis amigas Suly y Lilibet,** a mis queridas "cruela" y "katrina", por soportar mi carácter y lo más importante, confiar siempre en mí, como yo en ellas, las quiero chicas.*

***A mis amistades del 92105...**Yensy, Lilian, Madonna, Amal, Yamilka, Arianna, Aliané, Alina, por ser mi pequeña familia, compartir horas sin sueño y enseñarme que cada una tiene algo especial.*

***A mi compañero de tesis Andiel,** gracias por siempre decirme "todo está bien, hay tiempo", cuando en realidad no lo teníamos y estábamos que no podíamos más, tu apoyo fue de verdad importante.*

***A mi tutor Coca,** gracias por confiar en nosotros y decirnos siempre: "hay tiempo, tranquilos".*

***A mi co-tutor,** gracias por tu ayuda.*

***A los que de una u otra forma me ayudaron con la realización de esta tesis...**especialmente a Ramón, William y Leonel.*

***A todos, un millón de gracias...***

## **Andiel**

*Llegó la hora de despedirse de la universidad y la vida de universitario luego de tantos años de sacrificio y estudio me gustaría agradecer a todas aquellas personas que de una forma u otra fue parte de mi vida en estos años:*

***A mi abuelo** que en paz descansa, donde quiera que estés gracias por todo el amor que me brindaste, gracias por cada consejo que hoy me sirven para ser una mejor persona gracias a ti.*

***A mi madrina Susana** ya en la otra vida gracias por quererme como tu hijo.*

***A mi madre y a mi padre** por apoyarme, aconsejarme y ayudarme siempre en cada paso que he dado en estos años, por ser incondicionales y amarme, gracias, ustedes son mi vida.*

***A mi hermanito Felitín**, ese pequeño loquito, gracias por querer a tu tata como lo haces.*

***A mis dos mamis o abuelas** por la preocupación y atención que me han brindado siempre.*

***A todos mis primos** a los que quiero como hermanos.*

***A toda mi familia** en general les agradezco por brindarme su apoyo y amor incondicional.*

***A mi compañera de tesis Lislien** por todo lo que aprendimos y compartimos juntos. Por luchar juntos para sacar adelante la tesis, que tanto trabajo nos dió.*

***A mi tutor** quien siempre estuvo ahí para nosotros, apoyándonos, guiándonos, dándonos aliento para llegar a la meta.*

***A mi co-tutor** por su ayuda.*

## Dedicatoria

---

*Esta tesis se la dedico a las personas más importantes de mi vida:*

*A mi **abuelita Luisa**, por ser tan maravillosa y estar siempre que la necesito, a **mi mamá** por tanto amor y dedicación, a mi **tío Félix**, por ser mi "papá" en todos estos años, a **mis tías, Marle y Martha**, por ser mis segundas "mamis", apoyarme siempre y confiar en mí; a **mi padrastro Jesús**, por enseñarme el significado de ser independiente, a **mi hermanita Sianis**, por ser la luz de mis ojos y a mi **novio Yaxniel**, por apoyarme siempre y ser mi mejor amigo...*

**Lislien**

*Esta tesis se la dedico con todo mi corazón a las personas más importantes y especiales de mi vida: **A mi queridísimo abuelo** ausente pero presente en mi corazón, mi principal guía en los estudios, quien me impulso desde pequeño a dar lo mejor de mí, a estudiar, a ser buen hijo y un futuro profesional.*

***A mis padres** a quienes adoro con locura, gracias por su educación, por su amor .Hoy soy lo que soy gracias a ustedes, los amo.*

***A mi hermanito Felitín** para que el día de mañana siga mis pasos.*

***A toda mi familia**, de punta a cabo que para mí son lo primero ...*

**Andiel**

## Resumen

La visión por computador es una rama de la inteligencia artificial que desarrolla las tecnologías necesarias para simular todo el proceso de percepción de la visión humana. Una amplia gama de aplicaciones en esta rama requieren de la detección de objetos, sobre todo de personas, como información básica para alcanzar sus propósitos. Aunque existen muchas investigaciones en este sentido aún se trabaja en la integración de varias técnicas para reducir la cantidad de variables a analizar en el procesamiento de la información. Una de las técnicas que más ha impulsado el desarrollo de la visión por computador es la estereopsis, la cual brinda un mayor nivel de información de una escena, concretamente la profundidad de cada punto.

El presente trabajo propone una aplicación para representar mediante un mapa de oclusión la profundidad de las personas detectadas por uno de los algoritmos más reconocido en la bibliografía para el reconocimiento de características faciales.

**Palabras clave:** Visión por computador, detección, estereopsis, mapas de oclusión.

# Índice

Introducción.....	1
CAPÍTULO 1. Fundamentación Teórica.....	4
Introducción.....	4
1.1 Sistema de reconocimiento facial.....	4
1.2 Detección facial.....	5
1.3 Métodos de detección de rostros.....	5
1.3.1 Métodos basados en el conocimiento.....	6
1.3.2 Métodos basados en características invariantes.....	8
1.3.3 Métodos de coincidencia de plantillas.....	10
1.3.4 Métodos basados en apariencia.....	11
1.4 Percepción de Profundidad.....	12
1.4.1 Geometría epipolar.....	14
1.4.2 Rectificación de imágenes.....	15
1.4.3 Correspondencia de un punto en dos imágenes.....	17
1.5 Mapa de disparidad.....	17
1.6 Mapas de oclusión.....	18
1.7 Técnicas Ópticas.....	19
1.8 Calibración de un sistema de visión por computador.....	21
1.8.1 Tipos de calibración de cámara.....	23
1.8.2 Características de los métodos de calibración.....	23
1.8.3 Técnicas de calibración.....	23
1.9 Bibliotecas para el desarrollo de la visión por computador.....	24
1.9.1 OpenCV.....	24
1.9.2 ArToolkit.....	24
1.9.3 Touchless.....	25
1.9.4 Comparación entre las bibliotecas.....	25
1.10 Metodología de Desarrollo de Software.....	25
CAPÍTULO 2. Solución Propuesta.....	27



Introducción .....	27
2.1 Algoritmo de Viola y Jones.....	27
2.1.1 Imagen integral .....	28
2.1.2 Clasificador simple.....	29
2.1.3 Algoritmo AdaBoost .....	30
2.1.4 Detector mediante clasificadores en cascada .....	31
2.2 Información estéreo.....	32
2.2.1 Calibración de cámara.....	32
2.2.2 Selección del algoritmo de apareamiento estereoscópico .....	34
2.2.3 Cálculo de disparidad .....	34
2.2.4 Obtención del mapa de disparidad denso mediante la OpenCV. ....	35
2.3 Obtención de profundidad de los rostros en la imagen original .....	37
2.4 Obtención del mapa de oclusión.....	38
2.5 Recursos utilizados en el proceso de desarrollo .....	39
CAPÍTULO 3. Ingeniería del sistema.....	41
Introducción .....	41
3.1 Modelo de dominio .....	41
3.2 Personal relacionado con el Sistema .....	42
3.3 Exploración. Historia de Usuario.....	42
3.4 Lista de reserva del producto.....	44
3.5 Tareas de Ingeniería .....	46
3.6 Plan de Iteraciones.....	49
3.7 Modelo de Diseño .....	50
3.7.1 Diagrama de Clases .....	50
3.7.2 Tarjetas CRC .....	51
3.8 Patrones de Diseño.....	53
CAPÍTULO 4. Análisis de resultados .....	54
Introducción .....	54
4.1 Proceso de detección.....	54
4.2 Obtención de mapas de disparidad .....	55

---

4.3 Obtención de posiciones relativas .....	57
4.4 Pruebas de Aceptación .....	57
Conclusiones.....	60
Recomendaciones .....	61
Bibliografía.....	62
Glosario de términos .....	64

# Índice de Figuras

Figura 1. Esquema de las relaciones entre la visión por computadora y otras áreas afines (32).	1
Figura 2. Ejemplo de detección facial.	4
Figura 3. Cara típica utilizada en métodos basados en conocimiento (1).	6
Figura 4. Imagen original y sus correspondientes a menor resolución (3).	7
Figura 5. Algunas características faciales (1).	8
Figura 6. Ejemplo de un rectángulo rotado $45^\circ$ (5).	9
Figura 7. Ejemplo de imagen de 24 bits (dcha.) e imagen reducida a 8 colores (1).	9
Figura 8. Imagen de <i>board</i> obtenida en el formato RGB (izq.) y su correspondiente en el espacio (dcha.) (1).	10
Figura 9. El punto más cercano es enfocado por los ojos, y se proyecta en el centro de sus retinas, sin disparidad (13).	13
Figura 10. Geometría epipolar, el punto P, los centros ópticos O y O', las dos imágenes p y p' de P, todos se encuentran sobre el mismo plano (16).	15
Figura 11. Restricción epipolar: el conjunto de posibles apareamientos para el punto p se restringe a estar sobre la línea epipolar l' (16).	15
Figura 12. Un Par de imágenes rectificadas: los dos planos de la imagen $\Pi$ y $\Pi'$ son proyectados en un plano común $\Pi = \Pi'$ paralelo a la línea base. Las líneas epipolares l y l' asociadas con los puntos p y p' en las dos imágenes tiene la línea epipolar común $l = l'$ también paralela a la línea base y pasa a través de los puntos p y p' (16).	16
Figura 13. Triangulación para imágenes rectificadas: los rayos asociados con dos puntos p y p' en la misma línea de exploración se intersectan en el punto P. La profundidad de P relativa al sistema de coordenadas de la cámara izquierda es inversamente proporcional a la disparidad $d = u - u'$ (16).	17
Figura 14. Correspondencia de un punto en dos imágenes procedentes de un punto 3D común (19).	17
Figura 15. Mapa de disparidad obtenido con diámetro fijo de 7 píxeles (13).	18
Figura 16. Mapa de disparidad obtenido con diámetro variable (13).	18
Figura 17. Mapa de disparidad en sub-píxel con oclusiones marcadas en negro (13).	19
Figura 18. Clasificación de las técnicas ópticas (19).	20
Figura 19. Principales pasos en la visión estereoscópica (19).	21
Figura 20. Características utilizadas en el detector de Viola-Jones (1).	27
Figura 21. Representación de una imagen genérica conteniendo un modelo de característica (1).	28

Figura 22. El valor de la imagen integral en un punto (x,y) es la suma de los píxeles de arriba a la izquierda (1). .....	29
Figura 23. Cálculo del valor de la suma de los píxeles dentro de un área (1). .....	29
Figura 24. Algoritmo AdaBoost en pseudocódigo (1). .....	31
Figura 25. Estructura de clasificadores en cascada (1). .....	32
Figura 26. Ejemplos de patrones de calibración (19). .....	33
Figura 27. Par de imágenes estereoscópicas rectificadas. ....	34
Figura 28. Imagen obtenida aplicando el método <i>cvFindStereoCorrespondenceBM</i> . .....	36
Figura 29. Imagen obtenida aplicando el método <i>cvFindStereoCorrespondenceGC</i> . .....	36
Figura 30. Imagen con detección de rostros .....	37
Figura 31. Posiciones relativas de los rostros en la imagen. ....	37
Figura 32. Imagen binaria .....	38
Figura 33. Imagen con representación de profundidad. ....	39
Figura 34. Herramientas utilizadas en el proceso de desarrollo .....	40
Figura 35. Modelo de Dominio. ....	41
Figura 36. Diagrama de Clases del diseño. ....	51
Figura 37. Detección de rostros en tiempo de ejecución. ....	54
Figura 38. Imágenes con problemas de detección de rostro. ....	55
Figura 39. Primer par de imágenes analizadas. ....	55
Figura 40. Segundo par de imágenes analizadas. ....	56
Figura 41. Mapa de disparidad GC correspondiente al primer par de imágenes. ....	56
Figura 42. Mapa de disparidad GC correspondiente al segundo par de imágenes. ....	57
Figura 43. Posiciones relativas de los rostros correspondientes a la Figura 39. ....	57

# Índice de Tablas

Tabla 1. Personal relacionado con la aplicación. ....	42
Tabla 2. HU Detectar rostros dentro de la imagen. ....	43
Tabla 3. HU Detectar rostros en video. ....	43
Tabla 4. HU Obtener mapa de disparidad. ....	44
Tabla 5. HU Calcular posiciones relativas de los rostros detectados. ....	44
Tabla 6. Lista de reserva del producto. ....	45
Tabla 7. Tarea de Ingeniería Num.1 correspondiente a la HU Detectar rostros en la imagen. ....	46
Tabla 8. Tarea de Ingeniería Num.2 correspondiente a la HU Detectar rostros en la imagen. ....	46
Tabla 9. Tarea de Ingeniería Num.3 correspondiente a la HU Detectar rostros en la imagen. ....	47
Tabla 10. Tarea de Ingeniería Num.1 correspondiente a la HU Detectar rostros en video. ....	47
Tabla 11. Tarea de Ingeniería Num.1 correspondiente a la HU Obtener mapa de disparidad. ....	48
Tabla 12. Tarea de Ingeniería Num.1 correspondiente a la HU Calcular posiciones relativas de los rostros detectados. ....	48
Tabla 13. Tarea de Ingeniería Num.2 correspondiente a la HU Calcular posiciones relativas de los rostros detectados. ....	48
Tabla 14. Tarea de Ingeniería Num.3 correspondiente a la HU Calcular posiciones relativas de los rostros detectados. ....	49
Tabla 15. Plan de Iteraciones. ....	50
Tabla 16. Tarjeta CRC correspondiente a la clase "Form_Camara". ....	52
Tabla 17. Tarjeta CRC correspondiente a la clase "Form_Imagen". ....	52
Tabla 18. Tarjeta CRC correspondiente a la clase "MainWindow". ....	52
Tabla 19. Tarjeta CRC correspondiente a la clase "Persona". ....	53
Tabla 20. Caso de prueba de aceptación correspondiente a la HU 1. ....	58
Tabla 21. Caso de prueba de aceptación correspondiente a la HU 2. ....	59
Tabla 22. Caso de prueba de aceptación correspondiente a la HU 3. ....	59
Tabla 23. Caso de prueba de aceptación correspondiente a la HU 4. ....	59

## Introducción

La visión artificial o visión por computador es una disciplina cuyo objetivo es la extracción automática de información del mundo físico a partir de imágenes, la cual implica diversas áreas del conocimiento como la ingeniería, física, electrónica, neurobiología, entre otras (Figura 1). El continuo desarrollo de algoritmos, funciones y aplicaciones hace que sea una disciplina en continua evolución.

Se trata de un campo de estudio relativamente nuevo que comenzó a emerger aproximadamente a finales de la década de 1970. Cubre un amplio rango de temas que pertenecen también a otras disciplinas y no existe un problema estándar que defina qué debe resolver la visión artificial y cómo debe hacerlo. A lo largo de los últimos años se han desarrollado una gran variedad de técnicas y algoritmos que permiten resolver con mayor o menor grado de acierto diversos problemas de visión artificial.

Existen varios enfoques en los cuales se han centrado las investigaciones en este ámbito, la identificación de rostros a partir de una base de datos de rostros, el reconocimiento de expresiones faciales, el movimiento de los labios en videos con el objetivo de identificar expresiones o incluso palabras. Todos estos enfoques tienen como base la detección de rostros para lograr el análisis posterior de los mismos.

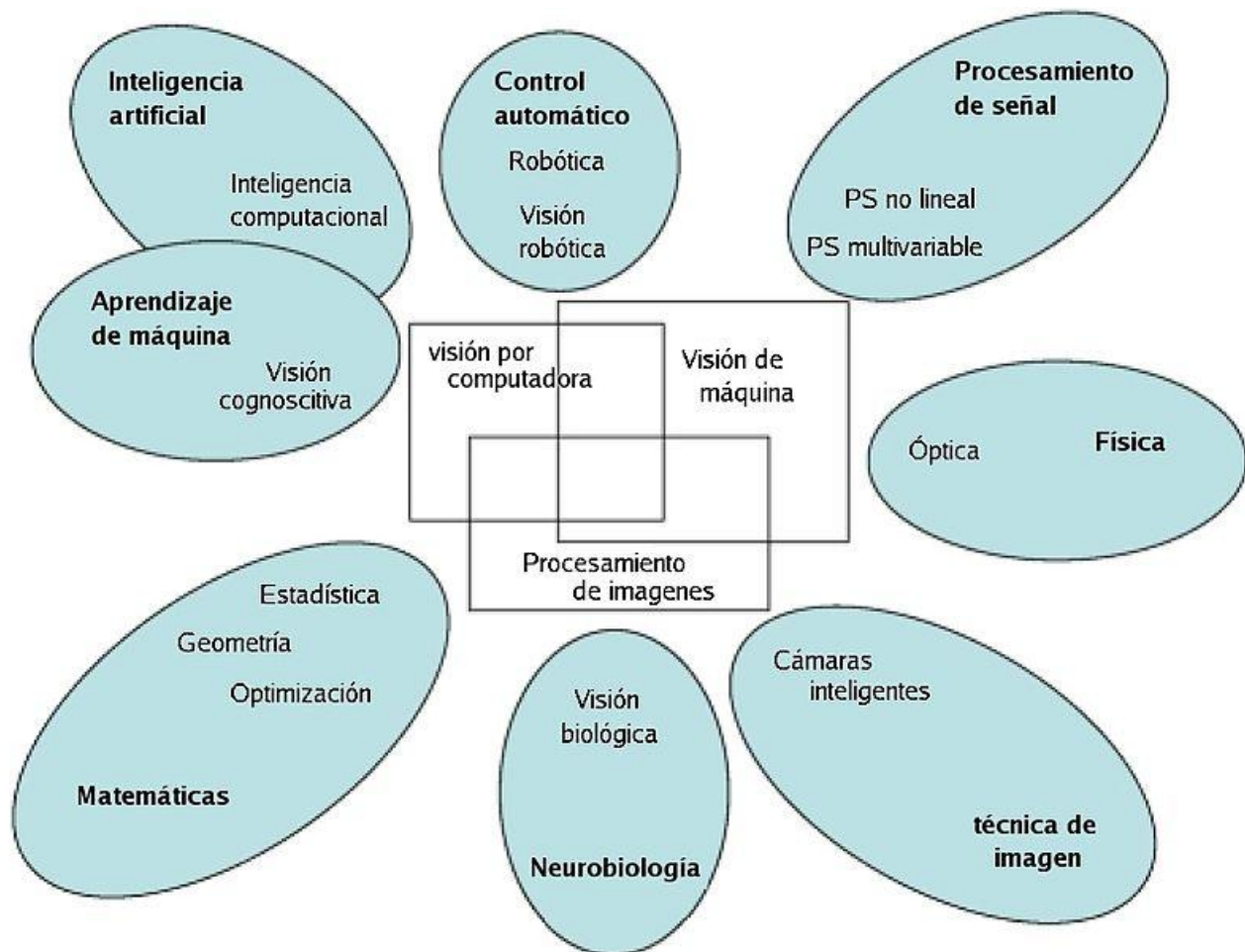


Figura 1. Esquema de las relaciones entre la visión por computadora y otras áreas afines (32).

Dentro de las aplicaciones más importantes de esta rama se encuentran las del campo de la medicina para procesamiento médico de imágenes de ayuda en diagnósticos, la industria, en el campo militar y en la detección de enemigos o en la guía de misiles, para dirigir vehículos de manera automática; en el campo de la biometría para identificación de personas, siendo precisamente el reconocimiento de rostros una de las técnicas biométricas más importantes en los últimos años, además de tener evidentes ventajas sobre otras como el reconocimiento por huellas dactilares o reconocimiento del iris del ojo.

En la facultad 5 de la Universidad de las Ciencias Informáticas, en el Centro de Informática Industrial (CEDIN), se llevan a cabo investigaciones relacionadas con la estereopsis, concretamente se han obtenido resultados en aspectos como la realidad aumentada y la determinación de la pose del observador. Uno de los proyectos que se propone desarrollar requiere el reconocimiento gestos de personas, para lo cual se necesita detectar las personas en un entorno y además guardar la mayor cantidad de información posible, sobre todo la profundidad a la cual se encuentran de la cámara. Cuando existen varias personas en una escena es importante determinar la posición relativa de cada una de ellas a la cámara, de esta forma se pudiera conocer el tamaño de las personas o determinadas poses o gestos. En la actualidad se utiliza la estereopsis con este objetivo, es decir, los mapas de disparidad obtenidos a partir de las imágenes tomadas simultáneamente desde dos cámaras, los cuales indican la distancia a la cámara de cada punto de la imagen. Sin embargo no se cuenta con una aplicación que permita detectar rostros y almacenar de una forma adecuada su posición relativa, incluyendo su profundidad, concretamente saber cuáles de estos están más alejados o más cercanos a la cámara.

Por lo antes expuesto queda identificado como **problema científico**:

- ¿Cómo representar la profundidad de personas a partir de la detección de rostros?

Se conoce que los mapas de oclusión han sido utilizados para saber si un objeto está ocultando a otro en una escena, por lo que se ha propuesto como **objetivo general**:

- Desarrollar una aplicación que permita detectar personas para representar su profundidad en la escena mediante un mapa de oclusión.

A partir del problema planteado, se toma como **objeto de estudio** la detección de personas y como **campo de acción** la representación de la profundidad de personas en una escena.

La idea a **defender de esta investigación** es: Con la utilización de los mapas de profundidad y a partir de una detección inicial de rostros se obtendrá un mapa de oclusión que guarde la profundidad de cada persona en la escena.

Para dar cumplimiento al objetivo planteado se han tomado las siguientes **tareas investigativas**:

- Elaboración del marco teórico de la investigación respondiendo al estado del arte actual del tema a investigar.
- Selección de la biblioteca a utilizar para la detección de rostros.
- Determinación de la imagen de disparidad a partir de la información estéreo.
- Representación de las posiciones relativas de los rostros detectados en la escena.

- Creación de mapas de oclusión en la detección de rostros.

Para todo el proceso de investigación y elaboración de este trabajo se tomó en cuenta la utilización de varios métodos científicos de investigación como:

- **Histórico – Lógico:** Mediante este método teórico se analizará la evolución y desarrollo de la visión por computador, el reconocimiento de personas y la detección de rostros, así como sus elementos más importantes.
- **Analítico – Sintético:** Mediante este método teórico se analizarán las teorías, documentos e información, permitiendo la extracción de los elementos más importantes que se relacionan con la visión por computador y el reconocimiento de personas.
- **Observación:** Mediante este método empírico se observarán los resultados obtenidos en la caracterización e identificación de los principales algoritmos utilizados, para poder decidir luego cuál o cuáles serán más adecuados.
- **Consulta de fuentes de información:** Método empírico que permite tener acceso a la información disponible sobre las bibliotecas para el desarrollo de la visión por computador y sus funcionalidades.

La estructura del documento está definida de la siguiente manera:

**Capítulo 1.** Fundamentación teórica: Se exponen los conceptos fundamentales relacionados con el tema de investigación y se presentan las bases teóricas fundamentales de las técnicas y algoritmos para la detección de personas existentes en la actualidad.

**Capítulo 2.** Solución Propuesta: Se expone la propuesta de solución al problema así como los métodos, técnicas, bibliotecas y herramientas utilizados.

**Capítulo 3.** Ingeniería del sistema: Se exponen el análisis, diseño y construcción de la solución propuesta.

**Capítulo 4.** Análisis de resultados: Se exponen las pruebas de precisión y aceptación realizadas.

Finalmente se presentan las conclusiones y recomendaciones obtenidas luego de concluir todo el proceso de investigación. Además se elaboró un glosario de términos con el objetivo de facilitar la comprensión del lenguaje usado.



# CAPÍTULO 1. Fundamentación Teórica

## Introducción

En este capítulo se abordarán los principales elementos teóricos que conforman tanto los sistemas como las técnicas de visión por computadora, lo cual permite obtener información a partir de imágenes bidimensionales. Se presta especial atención al reconocimiento de rostros, así como las técnicas para su detección y seguimiento, haciendo énfasis en la detección, utilizando para la misma los mapas de disparidad y los mapas de oclusión.

### 1.1 Sistema de reconocimiento facial

Los sistemas de reconocimiento facial son una aplicación dirigida por ordenador que identifica automáticamente a una persona en una imagen digital. Su objetivo es determinar qué parte o región de una fotografía es una cara y distinguirla mediante algún método de marcado, del resto de la imagen. Se puede englobar en la disciplina de detección de clases de objetos, en la que se buscan localizaciones y tamaños de objetos que pertenecen a una clase dada, como por ejemplo señales de tráfico, personas, códigos, figuras geométricas, etc.

La detección de regiones faciales está muy relacionada con la biometría, en cuanto que esta última utiliza detectores de caras para posteriormente extraer, procesar sus rasgos y obtener un resultado concreto como puede ser averiguar el nombre y apellidos de la persona que se haya situada enfrente de una cámara.

Hay varios factores que pueden afectar a los resultados de un sistema de detección de caras si este no es lo suficientemente sofisticado. Uno de ellos es la posición de esta respecto al plano de grabación ya que puede presentarse una cara muy cercana al borde o parcialmente oculta. Se dificulta asimismo el proceso de localización si se “viste” la cara con objetos que la tapan parcialmente como sombreros, pañuelos, gafas, barba, etc. (Figura 2). También las condiciones de iluminación o las características de la cámara, e incluso el cómo está configurada, pueden afectar notablemente a los resultados si no se han estudiado con la suficiente profundidad.



Figura 2. Ejemplo de detección facial.

## 1.2 Detección facial

Las imágenes que contienen caras son esenciales en muchos sistemas inteligentes de visión y muchos esfuerzos en el procesado facial incluyen el reconocimiento facial, el seguimiento facial, la estimación de la pose y el reconocimiento de expresiones. Sin embargo, muchos métodos asumen que las caras de una imagen se encuentran identificadas y localizadas. Para construir sistemas completamente automáticos, que analicen la información contenida en imágenes de caras, se requieren sistemas de detección facial robusta y eficiente.

Dada una imagen el objetivo del sistema de detección es identificar todas las regiones que contienen una cara sin importar su pose, orientación o condiciones de luz. Este problema presenta una gran dificultad ya que las caras son objetos diferentes según el sujeto y variables en tamaño, forma, color y textura (1).

- **Pose.** El ángulo de la cámara al capturar las imágenes puede variar rotando una cara hasta 180° desde una posición frontal dando lugar a poses frontales, de perfil, rotadas 45°, tomadas desde arriba o desde abajo. La orientación de la imagen también puede producir oclusiones en algunas características faciales como la nariz o los ojos.
- **Presencia de componentes estructurales.** La presencia de características faciales como barba, bigote y gafas influyen en el rendimiento del sistema y presentan un grado de variabilidad bastante alto.
- **Expresión facial.** La apariencia de las caras está directamente afectada por la expresión facial de la persona. En algunas aplicaciones biométricas se rechazan imágenes que muestren alguna expresión.
- **Oclusión.** Parte o la totalidad de una cara puede verse comprometida al ser tapada por diversos objetos o por la pose del usuario. En una imagen de un grupo de personas algunas caras pueden estar parcialmente tapadas por otras.
- **Orientación.** La rotación de una imagen afecta la posible localización de caras.
- **Condiciones de captura.** Cuando una imagen es capturada, factores como la iluminación y las características de la cámara afectarán la respuesta del sistema en el posterior procesado.

## 1.3 Métodos de detección de rostros

Solamente un proceso de reconocimiento basado en una cara compuesta de características puras puede ser denominado reconocimiento facial. La localización de dichas características, el proceso de detección facial, puede ser llevada a cabo por muchos métodos que se clasificarán en cuatro categorías, desde métodos basados en el desarrollo de reglas basadas en conocimientos previos de características, a métodos de búsqueda de características faciales invariantes (1):

- I. **Métodos basados en conocimiento.** Las reglas en estos métodos están basadas en el conocimiento humano sobre las características que definen una cara. La mayoría de las reglas utilizan la relación entre características. Estos métodos están designados principalmente para la localización facial.
- II. **Aproximaciones de características invariantes.** Estos algoritmos tienen por objetivo encontrar características estructurales que existen incluso cuando varían la pose, el punto de vista, las condiciones de luz, etc. Estos métodos están designados principalmente para la localización facial.

- III. **Métodos de coincidencia de plantillas.** A partir de un conjunto de muestras se construye un patrón facial estándar. La relación entre la imagen de muestra y el patrón definido es observado y utilizado para hacer una deducción. Estos métodos están designados tanto para la localización como para la detección facial.
- IV. **Métodos basados en apariencia.** Son similares a los anteriores, pero a diferencia de ellos las plantillas son obtenidas a partir de un conjunto de imágenes de entrenamiento que deberían capturar la variabilidad representativa de la apariencia de una cara. Estos métodos se usan principalmente para la detección facial.

### 1.3.1 Métodos basados en el conocimiento

El desarrollo de un sistema de detección facial basado en estos métodos implica que una serie de reglas son definidas antes de la implementación del sistema. Estas reglas se derivan del conocimiento del investigador sobre las caras humanas (Figura 3). En general estos métodos carecen de todo entrenamiento. El sistema se resume a lo que ha sido definido por el investigador que actúa como un límite sobre la efectividad de la detección.

Es fácil descubrir reglas que describan las características de una cara. Por ejemplo, una cara suele aparecer en una imagen con dos ojos simétricos entre sí, una nariz y una boca. Las relaciones entre características pueden ser representadas por sus distancias relativas y posiciones.

Uno de los problemas de estos métodos es la dificultad que se presenta al intentar traducir el conocimiento humano a reglas bien definidas. Existen dos tipos de reglas: estrictas, que dan lugar a una baja tasa de detección; o laxas, que producen una alta tasa de falsas detecciones.

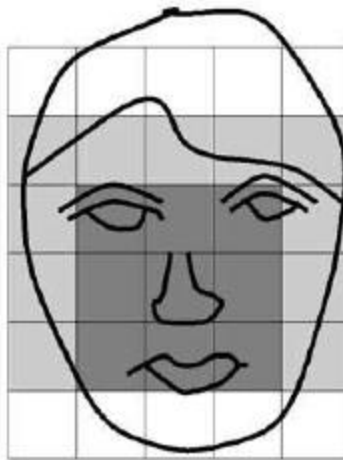


Figura 3. Cara típica utilizada en métodos basados en conocimiento (1).

#### Yang y Huang

Yang y Huang utilizaron un método basado en conocimiento jerárquico para detectar caras (2). Su sistema consiste en 3 niveles de reglas. En el nivel más alto todas las posibles caras son encontradas mediante un escaneado con una sub-ventana sobre la imagen y la aplicación de un conjunto de reglas en cada localización. En este primer nivel las reglas son descripciones generales de lo que es una cara y según se

van moviendo a otros niveles las reglas son más estrictas. Se crea una jerarquía de imágenes de multi-resolución mediante el cálculo de promedios y sub-muestreo como se ve en la Figura 4.



Figura 4. Imagen original y sus correspondientes a menor resolución (3).

### Tang, Kawato, Ohya y Nakatsu

La técnica presentada por estos investigadores (4) basa la detección en dos factores clave: distribuciones Gaussianas del color de la piel y el pelo, y diferenciación entre datos de piel irrelevantes utilizando una “curva de decisión” que localice la línea del pelo en una cara humana. En general, el método de detección puede dividirse en dos estados separados, cada uno de los cuales genera imágenes intermedias y que producirá una curva de decisión en el estado final.

- I. Generación de una distribución del color de la piel produciendo una imagen binaria.
- II. Generación de una distribución del color del pelo produciendo otra imagen binaria.
- III. Cálculo de una curva de decisión ofrecida por las imágenes binarias previas.

El primer estado se basa en un modelo de color de piel que se estima a partir de una distribución bidimensional Gaussiana cuyo propósito es la diferenciación respecto a un “fondo complejo”. El método se basa en la regla de que el color de piel está limitado a un rango de posibles colores RGB. En el primer estado dentro de la generación de la distribución de colores, se produce una imagen que contiene solamente los píxeles de la piel pintados. El cálculo de la curva de detección se produce en el siguiente estado del método y se basa en un modelo de pelo, basado de nuevo en una distribución Gaussiana en 2D. La aplicación del modelo de pelo a la imagen original también va a proveer una imagen secundaria que contenga pintados solamente los píxeles de la región del pelo.

Antes del cálculo de una curva de decisión las imágenes de muestra de los modelos de piel se observan y se realiza un pre-procesado para eliminar los elementos de la piel que no son representativos en una cara. Esto se hace mediante un escaneo de la imagen tanto en horizontal como en vertical y un reajuste de carreras de píxeles coloreados a cero si están por debajo de un umbral predeterminado. Esto producirá otra imagen y se estimará una curva de decisión a partir de ella.

Esta técnica es un buen ejemplo de un sistema basado en conocimiento, cuyo método se basa tanto en las distribuciones de color del pelo como de la piel y la presencia de una línea de pelo. Aunque esta técnica es bastante estricta, su aplicación ha demostrado que es bastante efectiva en detección facial de imágenes frontales, y también si la orientación facial ha sido alterada. La restricción obvia de este método es que el individuo debe tener pelo.

### 1.3.2 Métodos basados en características invariantes

Localizar en una imagen características útiles para la clasificación es una tarea compleja. A diferencia de los métodos basados en conocimiento, en este caso se buscan características invariantes para la detección facial. Se ha observado que los seres humanos pueden detectar caras y objetos sin esfuerzo en diferentes poses y condiciones de luz y por eso deben existir propiedades o características que sean invariantes sobre los conjuntos que se nos presentan. A partir de esta presunción numerosos métodos han sido propuestos para detectar características faciales y deducir si se da o no la presencia de una cara. Las características faciales (Figura 5) como cejas, ojos, nariz, boca, línea de pelo son extraídas comúnmente utilizando detectores de borde. Basándose en las características extraídas se construye un modelo estadístico que describa sus relaciones y verifique la existencia de una cara. Una limitación con estos algoritmos surge de la modificación de las características debido a la luminosidad, ruido, oclusiones, etc.

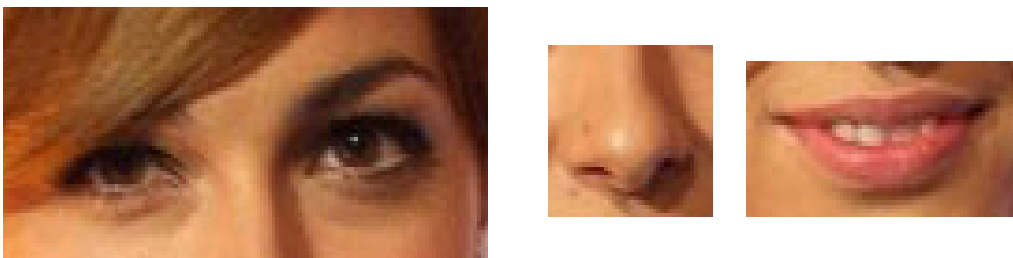


Figura 5. Algunas características faciales (1).

#### Lienhart y Maydt

Lienhart y Maydt (5) propusieron una mejora al algoritmo de Viola-Jones, introduciendo una ampliación del conjunto de características de *Haar* utilizadas. La técnica propuesta da lugar a resultados de detección más efectivos y un error de clasificación menor a través del uso de un conjunto de características modificado, permutaciones del conjunto de *Haar* y la adopción del algoritmo “*Gentle Adaboost*” para la selección de características.

Lienhart y Maydt mencionaron que los sistemas basados en características, al contrario que los basados en píxeles, reducen la variabilidad intra-clase. También establecieron que al combinar las características con un método de selección como *Gentle Adaboost* la capacidad de selección puede aumentar. Las características utilizadas por Viola-Jones se modificaron rotando cada una de ellas 45° tanto en sentido positivo como negativo (Figura 6) y algunas fueron elegidas para la detección. Se vio que se produjo un descenso de aproximadamente el 10% en las tasas de falsa alarma frente a lo presentado por Viola-Jones.

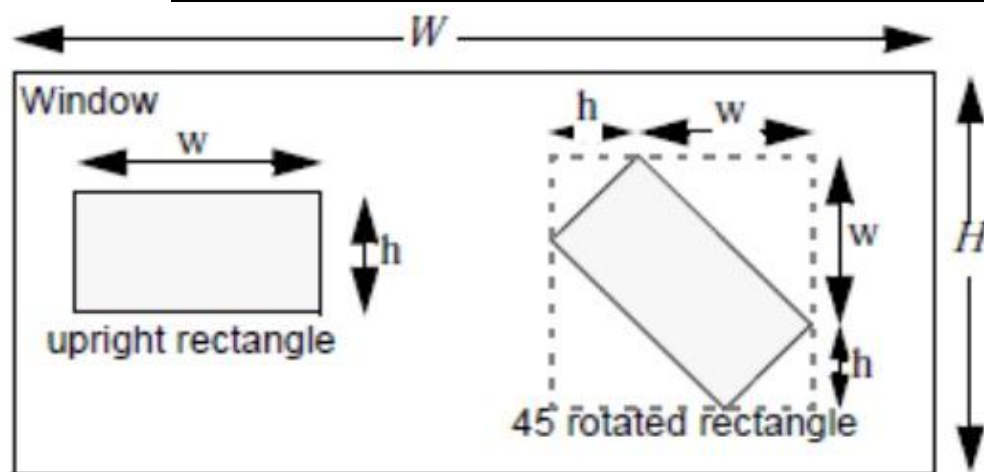


Figura 6. Ejemplo de un rectángulo rotado 45° (5).

La técnica también evaluó una gran cantidad de algoritmos de *boosting* e implementó el *Gentle Adaboost*, que en comparación a los otros se concentra menos en los ejemplos difíciles de definir. Esto es bastante diferente al proceso de AdaBoost presentado por Freund y Schapire (6).

### Garcia y Tziritas

El color y la textura también son considerados como un conjunto importante en una cara. Garcia y Tziritas presentaron una técnica que utiliza regiones de color de piel cuantificadas y un análisis de paquetes de *wavelets* (7). Su método tiene dos etapas principales: la detección del color de piel y el análisis efectivo de la textura utilizando una aplicación de análisis mediante filtros de paquetes *wavelets*.



Figura 7. Ejemplo de imagen de 24 bits (dcha.) e imagen reducida a 8 colores (1).

En el primer estado se realiza una cuantificación del color de la imagen y posteriormente se detectan las regiones con color de piel. La cuantificación ayuda a mejorar la segmentación de los colores de piel homogeneizando las regiones de la imagen. Esta cuantificación se consigue a través de la cuantificación de vector y reduce los colores que aparecen en la imagen a un conjunto menor (Figura 7). Acto seguido se produce la segmentación del color de la piel en la imagen obtenida. El método utiliza dos modelos para la representación del color: YCbCr y HSV.



La detección de posibles regiones faciales se consigue mediante la unión de regiones homogéneas de color de piel. El algoritmo de unión calcula un gráfico de adyacencia entre regiones, donde cada nodo es una región de color de piel de la imagen cuantificada, y conecta o no las regiones basándose en la similitud de los colores.

En la última etapa de esta técnica se produce la clasificación haciendo uso del análisis de paquetes de *wavelets* de aquellas regiones que sean posibles caras. Eligieron el uso de *wavelets* debido a su flexibilidad de aplicación y velocidad de cálculo. Se produce una descomposición por paquetes de *wavelets* en el plano de la intensidad de la imagen y se produce un vector de características de coeficientes de *wavelets*. La deducción se realiza aplicando un cálculo de distancias sobre los vectores de características extraídos de las áreas de color de piel segmentadas. Estos vectores caracterizan la textura facial.

### 1.3.3 Métodos de coincidencia de plantillas

En los métodos de coincidencia de plantillas un patrón facial estándar (normalmente de una cara en pose frontal) se predefine manualmente o con parámetros mediante una función. Dada una imagen de entrada, se calculan unos valores de correlación utilizando los patrones estándar para el contorno facial, los ojos, la nariz y la boca independientemente. La existencia de una cara se determina en base a los valores de correlación. Estos métodos tienen como ventaja una implementación sencilla, sin embargo se ha probado que son ineficaces para la tarea de la detección facial ya que no pueden tratar efectivamente la variación en escala, pose y forma de la cara.

La aplicación de una plantilla es similar a los métodos basados en conocimiento, ya que el conocimiento de las características faciales o bien puede ser aprendido y construir así una plantilla dinámica, o bien puede ser predefinido. La aplicación de una plantilla podría realizarse a diferentes escalas debido a la simpleza de muchas plantillas que se relacionan simplemente con la simetría de las características faciales y con las distancias relativas. Sin embargo, como se ha dicho, utilizar solamente estos métodos no es muy efectivo con lo que a menudo suelen utilizarse junto a otras técnicas basadas en características invariantes. Los métodos descritos a continuación se centran sólo en aquellas técnicas donde la coincidencia de plantillas (Figura 8) es la base para la detección.

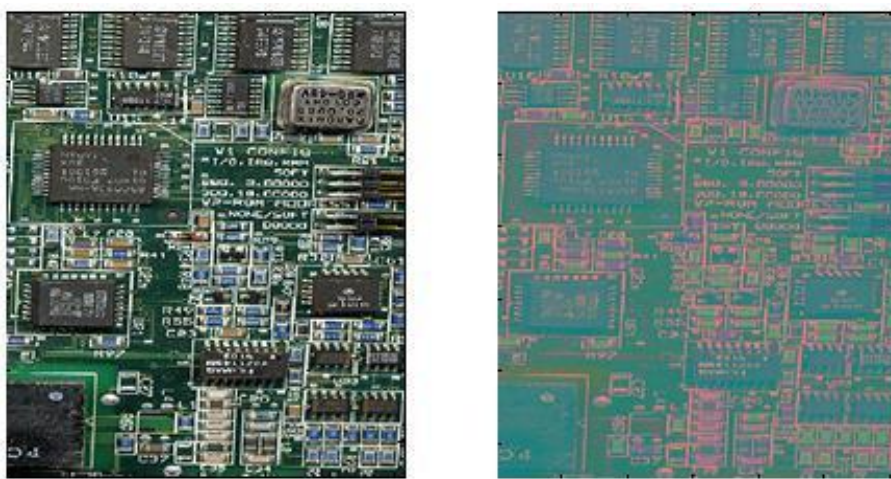


Figura 8. Imagen de *board* obtenida en el formato RGB (izq.) y su correspondiente en el espacio (dcha.) (1).

## Yuille, Hallinan, Cohen

En este método se propone detectar y describir características faciales utilizando plantillas deformables que modelen esas características y que encajen en un modelo elástico *a priori* (por ejemplo los ojos) (8). En esta técnica las características faciales se describen mediante plantillas, que han sido especificadas utilizando parámetros que posibilitan el conocimiento previo sobre la forma esperada de las características para guiar el proceso de detección. Estas plantillas son lo suficientemente flexibles como para poder cambiar su tamaño, y otros parámetros, para ajustarse a los datos. Se define una función de energía para unir bordes, picos y valles en la imagen de entrada con los correspondientes parámetros en la plantilla. El mejor ajuste del modelo elástico a la imagen se realiza minimizando la función de energía de los parámetros.

Una vez hecho esto los parámetros de la plantilla se actualizan mediante descenso por gradiente. Cambiar estos parámetros se corresponde con una alteración en la posición, orientación, tamaño y otras propiedades de la plantilla.

Aunque sus resultados experimentales hayan demostrado un buen rendimiento en el rastreo de características no rígidas, una desventaja de este método es que la plantilla debe ser inicializada en las cercanías del objeto de interés.

## Sinha

Esta técnica (9) utiliza un pequeño conjunto de características espaciales invariantes en una imagen para describir el espacio de los patrones faciales. Para designar lo invariante se basa en que, mientras que las variaciones en iluminación cambian el brillo individual de diferentes partes faciales (ojos, mejillas, frente) el brillo relativo de estas partes permanece aproximadamente invariable. Determinar estos ratios en unas cuantas regiones y guardar solamente su dirección (por ejemplo: ¿Es una región más brillante que otra?) proporciona una característica invariante bastante robusta.

### 1.3.4 Métodos basados en apariencia

Anteriormente los métodos de coincidencia de plantillas se basaban en una definición de las plantillas por parte del investigador. Sin embargo, en los métodos basados en apariencia se aprende de ejemplos en imágenes. En general estos métodos se basan en técnicas de análisis estadístico y aprendizaje automático para encontrar características relevantes de imágenes faciales o de imágenes que no contengan caras. Las características aprendidas se expresan como modelos de distribución o funciones discriminantes que son utilizadas para la detección facial. Muchas son las técnicas utilizadas: *eigenfaces*, métodos basados en distribuciones, redes neuronales, máquinas de vectores de soporte, clasificadores bayesianos, modelos de Markov, etc. Algunos ejemplos pasan a explicarse a continuación.

## Pham, Worring y Smeulders

En este método se presenta una red Bayesiana con una estructura en forma de bosque para resolver un problema de clasificación de una sola clase (10). Se eligen los clasificadores Bayesianos debido a la gran velocidad que estos pueden alcanzar. La técnica también hace uso de *bagging* para generar un clasificador agregado ya que proporciona una manera natural de resolver los problemas de clasificación referidos a una sola clase.



El método de detección facial aquí referido se construye sobre una base de resolución de 20 píxeles que se escala con un factor de 1.2 hasta que el tamaño de la imagen sea menor que el de la ventana escalada. La resolución de 20 píxeles fue elegida al considerarse por parte de los autores más que adecuada para contener las características de una cara humana y minimizar la relación entre la resolución de y el tiempo de clasificación. En el conjunto de datos la técnica presenta variaciones en iluminación, expresión, orientación y presencia o falta de componentes faciales como barba o gafas. La imagen también se somete a un pre-procesamiento lo cual implica una normalización del gradiente de iluminación y una ecualización de histograma. Esto reduce el efecto de las sombras y suaviza el contraste de la imagen.

Un clasificador agregado sirve como núcleo de la decisión del sistema. Está compuesto por tres clasificadores de redes Bayesianas elegidos a través de un proceso de *bagging*. La selección se basa en las soluciones observadas y los compromisos que interesen que se tomen. Por ejemplo, cuando el número de clasificadores aumenta la tasa de detección disminuye así como también lo hace la tasa de verdaderos positivos. Después de que la imagen fuese examinada y sus regiones etiquetadas como candidatas a ser caras o no, hay una etapa de post-procesamiento en la cual se seleccionan las regiones con los mayores valores de probabilidad y se marcan como caras.

### **Roth, Yang y Ahuja**

Roth, Yang y Ahuja (11) proponen una técnica diferente denominada SNOW (*Sparse Network of Winnows*) la cual utiliza una red formada por unidades lineales para definir el espacio de las características aprendidas. Al utilizar su método se da una relevancia particular a las tareas de clasificación que tengan grandes conjuntos de características. Dicho método muestra buenos resultados en la fase de aprendizaje en dichas circunstancias. Mediante el uso de características booleanas primitivas la técnica codifica tanto la intensidad, como los datos de la posición de los píxeles en la imagen de muestra.

Los nodos en la capa de entrada de la red representan relaciones simples entre las entradas y se utilizan como características de entrada. Cada unidad lineal se llama nodo objetivo y representa las relaciones de interés entre los ejemplos de entrada. La decisión en este sistema se hace mediante la unión de los nodos objetivo, en este caso sólo dos: cara y no cara. Dado un conjunto de relaciones, es decir tipos de características, que puedan existir en una imagen, cada imagen se asocia a uno de esos conjuntos de características presentes y SNOW propagará estas relaciones a un nodo objetivo y se tomará una decisión.

La aplicación de SNOW da lugar a detectar caras en múltiples poses, expresiones y diferentes niveles de iluminación. La aplicación del algoritmo *Winnow* en la red proporciona una manera de aprendizaje que se ajusta a dominios donde el espacio de características es grande y desconocido. El uso de la técnica se ha probado como una de las mejores implementaciones. Esto puede deberse a la aplicación del algoritmo SNOW y la red de decisión que se produce. La decisión es bastante robusta debido al uso del algoritmo *Winnow*.

## **1.4 Percepción de Profundidad**

A través de la visión binocular, somos capaces de interactuar en un mundo tridimensional al apreciar las distancias y volúmenes en el entorno que nos rodea. Nuestros ojos, debido a su separación, obtienen dos imágenes con pequeñas diferencias entre ellas, a lo que se denomina disparidad. Nuestro cerebro procesa las diferencias entre ambas imágenes y las interpreta de forma que percibimos la sensación de profundidad,

lejanía o cercanía de los objetos que nos rodean. Este proceso se denomina estereopsis. La distancia interpupilar más habitual es de 65 mm, pero puede variar desde los 45 a los 75 mm (12).

En la estereopsis intervienen diversos mecanismos:

- Cuando se observan objetos muy lejanos, los ejes ópticos de nuestros ojos son paralelos.
- Cuando se observa un objeto cercano, nuestros ojos giran para que los ejes ópticos estén alineados sobre él, es decir, converjan.

A su vez se produce la acomodación o enfoque para ver nítidamente el objeto. Este proceso conjunto se llama fusión. No todo el mundo tiene la misma capacidad de fusionar un par de imágenes en una sola tridimensional. Alrededor de un 5% de la población tiene problemas de fusión. La agudeza estereoscópica es la capacidad de discernir, mediante la estereopsis, detalles situados en planos diferentes y a una distancia mínima. Hay una distancia límite a partir de la cual no somos capaces de apreciar la separación de planos, y que varía de unas persona a otras. Así, la distancia límite a la que dejamos de percibir la sensación estereoscópica puede variar desde unos 60 metros hasta cientos de metros.

La Figura 9 ilustra una situación simplificada en dos dimensiones.

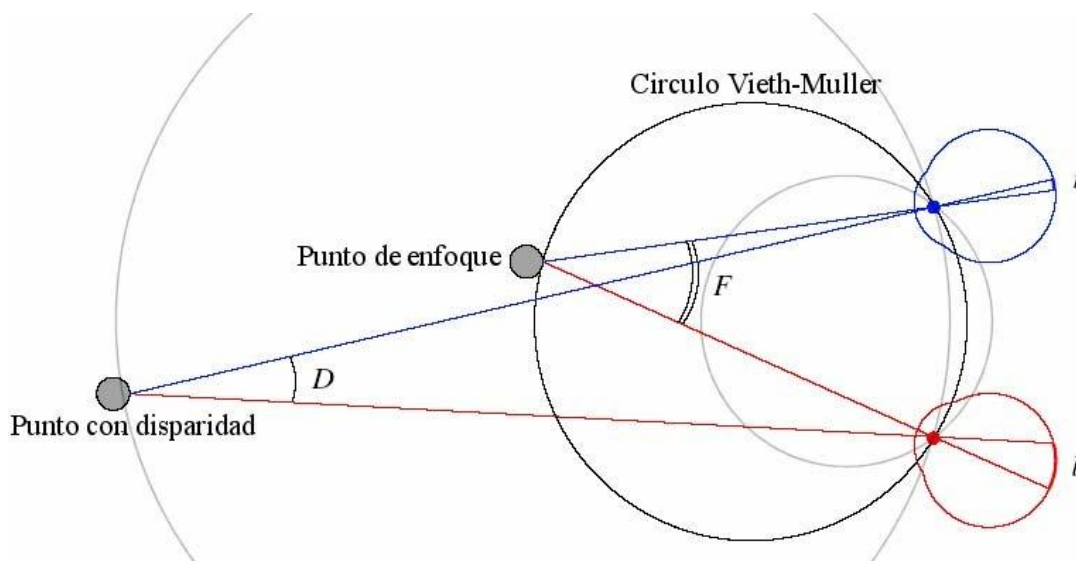


Figura 9. El punto más cercano es enfocado por los ojos, y se proyecta en el centro de sus retinas, sin disparidad (13).

Si  $l$  y  $r$  denotan el ángulo (en sentido antihorario) entre los planos de simetría vertical de los dos ojos y los dos rayos pasando a través de el mismo punto en la escena, entonces se denotan sus disparidades correspondientes como  $d = r - l$  y por lo tanto como se puede demostrar por trigonometría  $d = D - F$ , donde  $D$  es el ángulo entre estos rayos y  $F$  es el ángulo entre los rayos que pasan por el punto de enfoque. Los puntos con disparidad cero caen en el círculo *Vieth-Muller* que pasa por el punto enfocado y los centros focales de los ojos. Los puntos que se encuentran adentro del círculo tienen una disparidad positiva (o convergente), los que están fuera del círculo tienen una disparidad negativa (o divergente), y la posición de todos los puntos que tienen una disparidad dada  $d$ , forman,  $d$  como variable, un círculo que pasa a través de los dos centros focales. Esta propiedad es claramente suficiente para ordenar por orden de disparidad los puntos que están cerca del punto de enfoque.

Sin embargo también es claro que los ángulos entre el plano medio vertical de simetría de la cabeza y los dos rayos de enfoque se deben conocer para poder reconstruir la posición absoluta de los puntos en la escena.

El caso tridimensional es un poco más complicado, la posición para los puntos de disparidad cero se convierte en una superficie, *horopter*, pero la conclusión general es la misma, una posición absoluta requiere de los ángulos. Como fue demostrado por Wundt y Helmholtz (14) hay evidencia clara de que estos ángulos no pueden ser evaluados con precisión por nuestro sistema nervioso. Por lo tanto se puede argumentar razonablemente que la salida del estereopsis humana consiste en un mapa de profundidad relativa. Transmitiendo un orden parcial de profundidad ente los puntos (15). En este contexto el papel principal del movimiento de los ojos en la estereopsis es poder traer las imágenes dentro del área de fusión, un disco pequeño en el centro de la retina donde la fusión puede ocurrir (15) (se pueden percibir puntos para disparidades mucho más grandes pero aparecerán como imágenes dobles, un fenómeno conocido como diplopía).

Un factor que interviene directamente en la estereopsis es la separación interocular. A mayor separación entre los ojos, mayor es la distancia a la que se aprecia el efecto de relieve. Esto se aplica por ejemplo en los prismáticos, en los que, mediante prismas, se consigue una separación interocular efectiva mayor que la normal, con lo que se consigue apreciar en relieve objetos distantes que en condiciones normales no se tendría la capacidad de separar del entorno.

### 1.4.1 Geometría epipolar

La geometría epipolar describe las relaciones geométricas de las imágenes formadas en dos o más cámaras enfocadas en un mismo punto o polo. Los elementos más importantes de este sistema geométrico (Figura 10) son: El plano epipolar, formado por el polo (P) y los dos centros ópticos (O y O') de dos cámaras. Los epipolos (e y e') que son la imagen virtual del centro óptico (O' y O) de una cámara en otra. La línea base, la cual une los dos centros ópticos. Y las líneas epipolares (l y l'), formadas por las intersecciones del plano epipolar con los planos de las imágenes ( $\Pi$  y  $\Pi'$ ), esta une el epipolo con la imagen del punto observado (p, p') (16).

La línea epipolar es fundamental en la visión estereoscópica, pues una de las partes más difíciles en el análisis estereoscópico es establecer la correspondencia entre dos imágenes, apareamiento estéreo, decidiendo que punto en la imagen derecha corresponde a cual en la izquierda.

La restricción epipolar (Figura 11): "Si p y p' son imágenes del mismo punto P, entonces p' debe encontrarse en la línea epipolar l' asociada con p", nos permite reducir la búsqueda de la correspondencia estereoscópica, de dos dimensiones (toda la imagen) a una búsqueda en una dimensión sobre la línea epipolar (16).

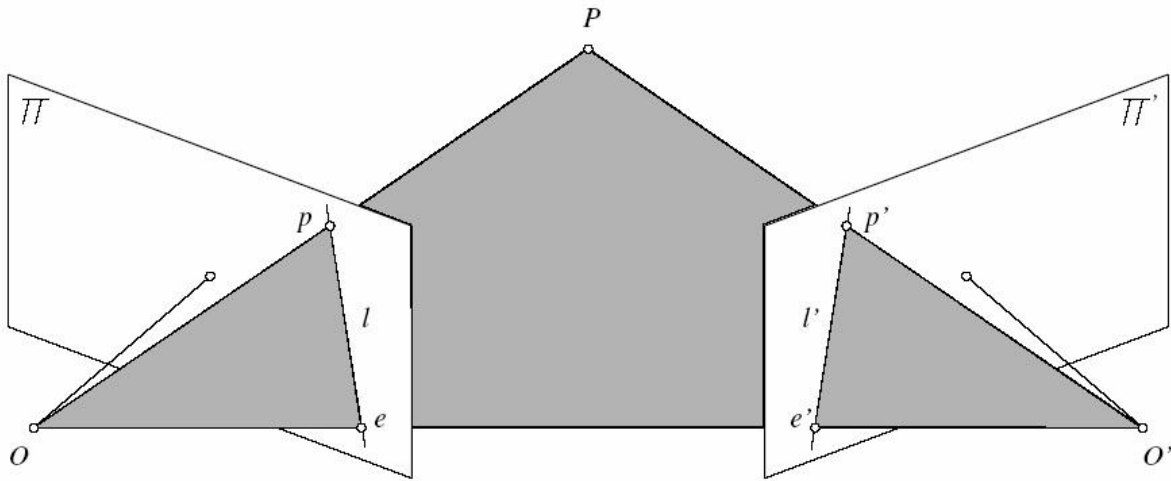


Figura 10. Geometría epipolar, el punto  $P$ , los centros ópticos  $O$  y  $O'$ , las dos imágenes  $p$  y  $p'$  de  $P$ , todos se encuentran sobre el mismo plano (16).

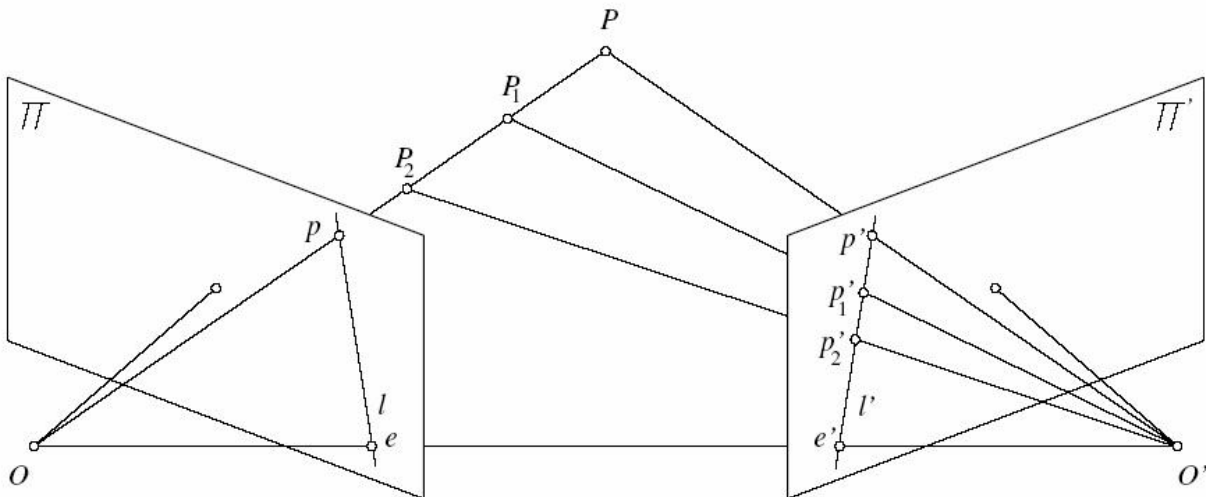


Figura 11. Restricción epipolar: el conjunto de posibles apareamientos para el punto  $p$  se restringe a estar sobre la línea epipolar  $l'$  (16).

### 1.4.2 Rectificación de imágenes.

Una forma de simplificar aún más los cálculos asociados con los algoritmos estereoscópicos es rectificar las imágenes, esto consiste en reemplazar las imágenes por sus proyecciones equivalentes sobre un plano común paralelo a la línea base (Figura 12).

Se proyecta la imagen, eligiendo un apropiado sistema de coordenadas, las líneas epipolares rectificadas son paralelas a la línea base y se convierten en una sola línea de exploración. Existen dos grados de libertad involucrados en la selección del plano de rectificación (17):

1. La distancia entre el plano y la línea base, este es irrelevante pues al modificarlo solo se cambia la escala de las imágenes rectificadas, su efecto se balancea fácilmente haciendo un escalamiento inverso.

- La dirección del plano rectificadado normal en el plano perpendicular a la línea base, la elección natural incluye escoger un plano paralelo a la línea donde las dos retinas originales se intersectan y minimizar la distorsión asociada con el proceso de proyección.

Para el caso de las imágenes rectificadas (Figura 13) dados dos puntos  $p$  y  $p'$  localizados en la misma línea de exploración de la imagen izquierda y derecha, con coordenadas  $(u, v)$  y  $(u', v')$ , la disparidad es dada como la diferencia  $d = u' - u$ . Si  $B$  es la distancia entre los centros ópticos, también llamada línea base, es fácil demostrar que la profundidad de  $P$  es  $z = -B / d$  (16).

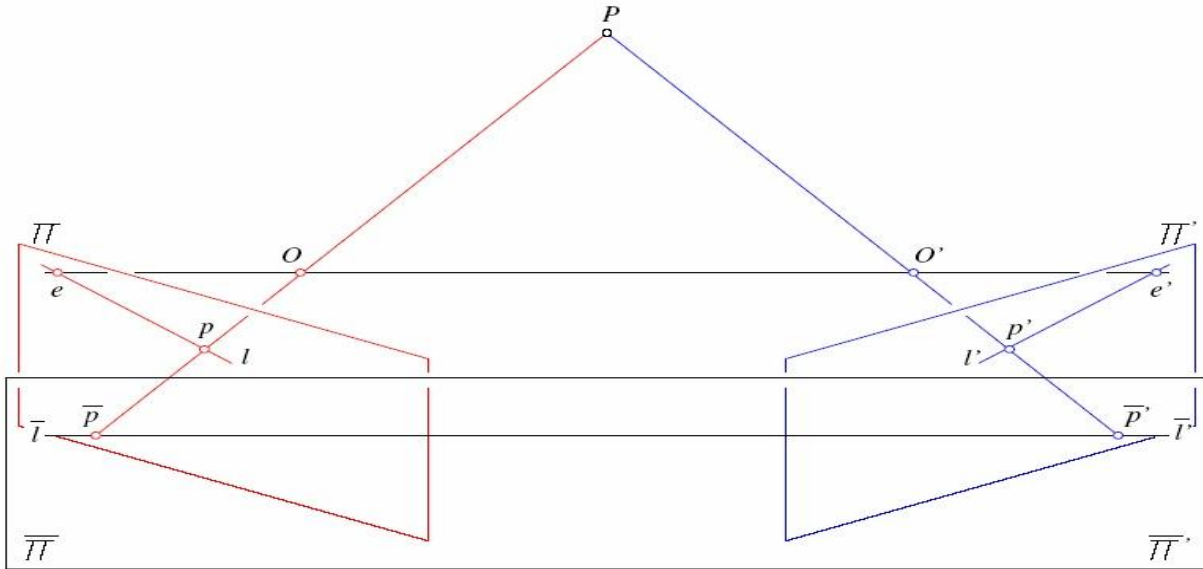


Figura 12. Un Par de imágenes rectificadas: los dos planos de la imagen  $\Pi$  y  $\Pi'$  son proyectados en un plano común  $\Pi = \Pi'$  paralelo a la línea base. Las líneas epipolares  $l$  y  $l'$  asociadas con los puntos  $p$  y  $p'$  en las dos imágenes tiene la línea epipolar común  $l = l'$  también paralela a la línea base y pasa a través de los puntos  $p$  y  $p'$  (16).

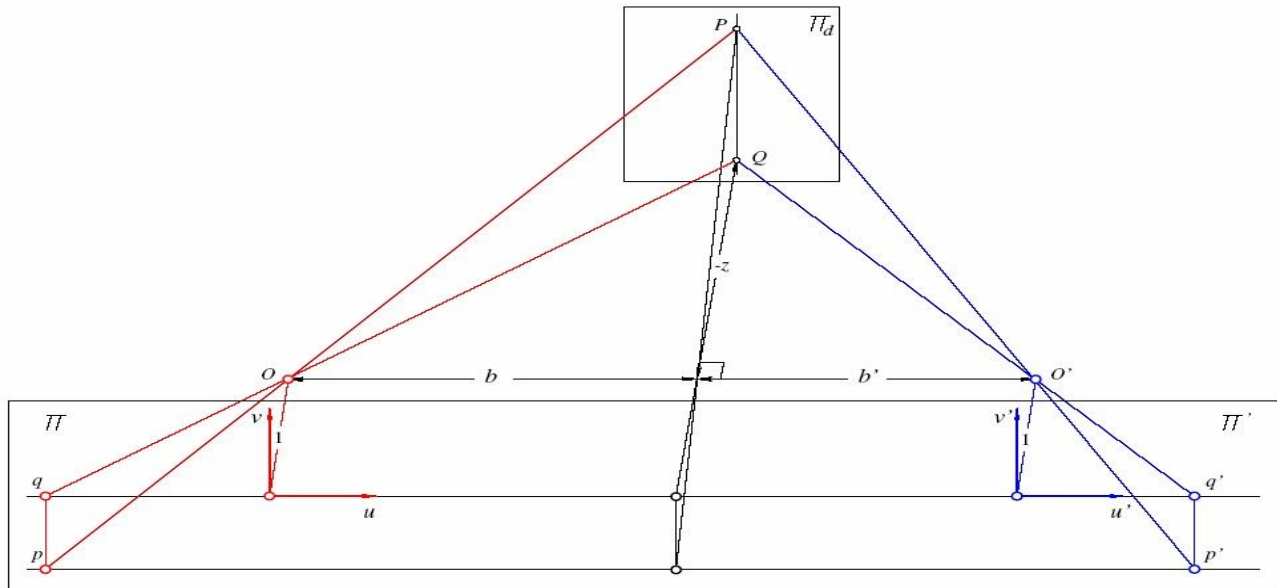


Figura 13. Triangulación para imágenes rectificadas: los rayos asociados con dos puntos  $p$  y  $p'$  en la misma línea de exploración se intersectan en el punto  $P$ . La profundidad de  $P$  relativa al sistema de coordenadas de la cámara izquierda es inversamente proporcional a la disparidad  $d = u - u'$  (16).

### 1.4.3 Correspondencia de un punto en dos imágenes.

Existen varios métodos para obtener la correspondencia, estos métodos se pueden dividir en locales y globales (18). Debido a que generalmente se obtienen falsas correspondencias, sobre todo en zonas donde las intensidades del punto son muy parecidas, o cuando los puntos en la imagen izquierda no aparecen en la imagen derecha y viceversa, se determinó realizar la correspondencia estéreo o la relación existente entre dos cámaras, manualmente. Cuando se parte de la correspondencia de puntos de dos imágenes comúnmente se emplea la matriz fundamental para obtener los parámetros de la cámara. Es posible estimar la matriz fundamental a partir de más de siete puntos de control, al usar esta matriz fundamental se puede obtener la matriz de proyección y realizar la proyección para la adquisición de información tridimensional en un espacio proyectivo. La solución se construirá en base a la correspondencia manual para la extracción de los parámetros extrínsecos de la cámara, tanto el vector de traslación como el de rotación. En la Figura 14 se muestra la correspondencia de puntos en dos vistas procedentes de un punto 3D común.

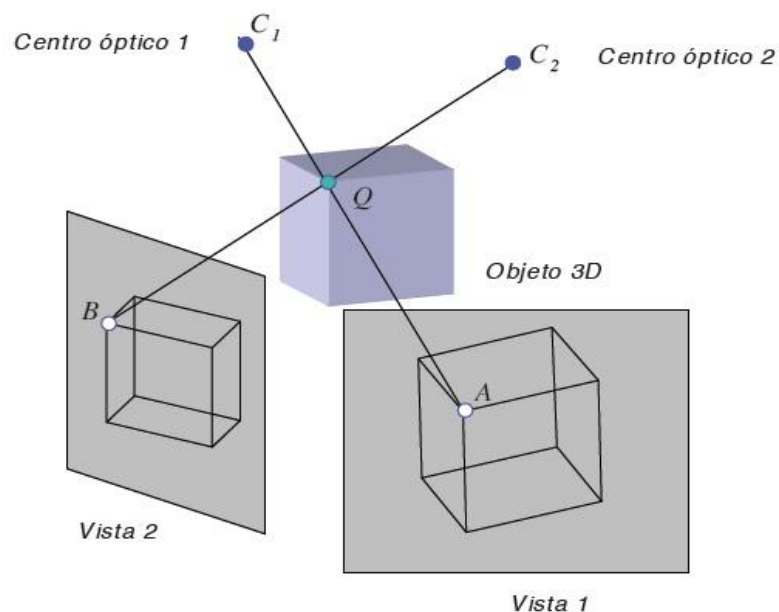


Figura 14. Correspondencia de un punto en dos imágenes procedentes de un punto 3D común (19).

## 1.5 Mapa de disparidad

Un mapa de disparidad es un arreglo bidimensional con las mismas dimensiones que la imagen de referencia, en donde cada localidad indica la distancia horizontal en píxeles a la cual se encuentra el mejor apareamiento encontrado en la otra imagen para la comparación estereoscópica. Se representa mediante una imagen donde la intensidad de tono en cada píxel indica a que distancia horizontal se encuentra el píxel correspondiente de una imagen en la otra (16) (Figura 15). El mapa de disparidad contendrá el valor de disparidad para cada píxel en la cual se encontró el mejor apareamiento. En la Figura 15 y Figura 16 los niveles de disparidad se indican en escala de grises de 256 tonos.



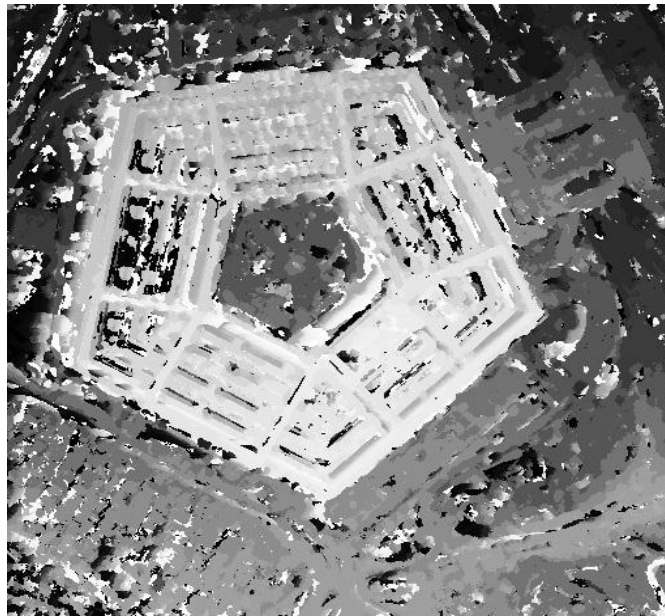


Figura 15. Mapa de disparidad obtenido con diámetro fijo de 7 píxeles (13).

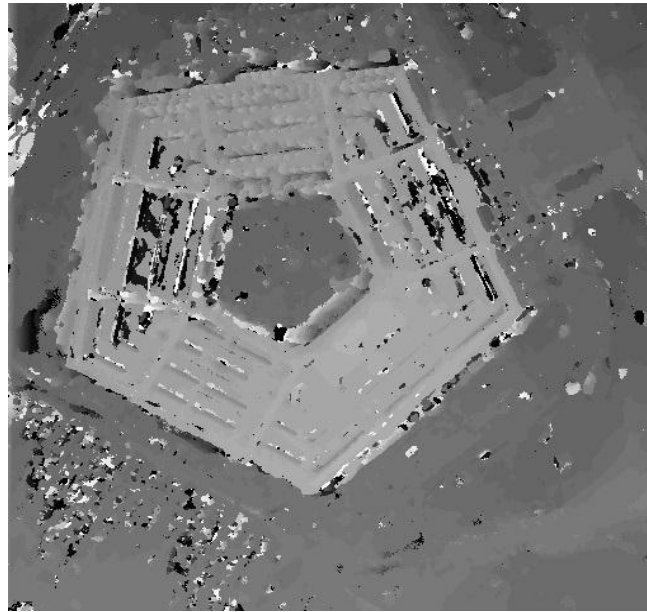


Figura 16. Mapa de disparidad obtenido con diámetro variable (13).

## 1.6 Mapas de oclusión

En un par de imágenes estereoscópicas existen detalles que se encuentran en una imagen pero no en la otra, pues se encuentran ocluidos por objetos cercanos debido a la diferencia de perspectiva entre ellas y a los cambios en profundidad. El costo de aparear una característica será muy alto si esta se encuentra ocluida en la otra imagen, pues no existirá similitud entre ninguno de los apareamientos dentro del rango de disparidad. Para determinar las oclusiones se toman todos los puntos del mapa de disparidad cuyo costo de oclusión sea mayor a un umbral  $[tho]$  y se les marca como ocluidos (Figura 17) (1).

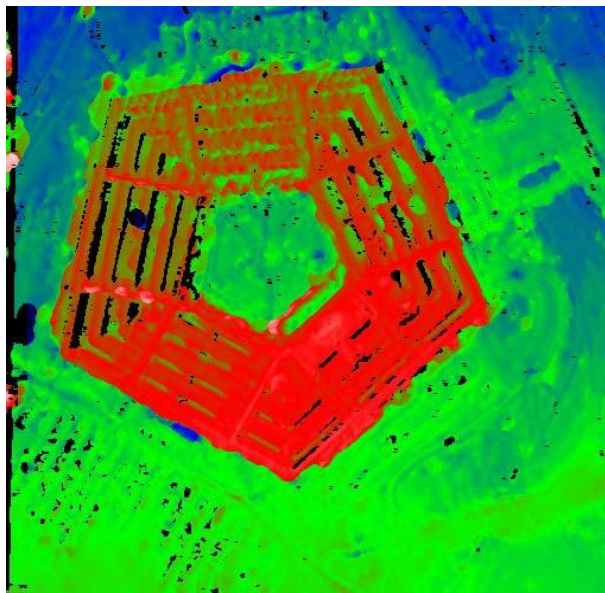


Figura 17. Mapa de disparidad en sub-píxel con oclusiones marcadas en negro (13).

Los mapas de oclusión permiten determinar si un objeto o persona está siendo ocultada por otra, requiere actualización en cada instante de tiempo y que el seguimiento se realice siguiendo un orden, por ejemplo comenzando siempre por la persona más cercana a la cámara, creando una imagen binaria donde cada punto inicialmente está en 0 y se pondrán en 1 los puntos que se determinen forman parte de una persona.

## 1.7 Técnicas Ópticas

Las técnicas ópticas se clasifican en dos grupos dependiendo de las características de las fuentes de luz utilizadas para iluminar la escena (Figura 18). Las técnicas pasivas no toman en cuenta la geometría de las fuentes de iluminación con respecto a la escena y al sensor a la hora de calcular las coordenadas tridimensionales de los objetos, por otra parte, las técnicas activas utilizan fuentes de luz con características específicas y su posición influye a la hora de determinar dichas coordenadas. Esta fuente de iluminación es conocida como fuente de luz activa ya que se hace necesario distinguirla del resto de iluminación de la escena.





Figura 18. Clasificación de las técnicas ópticas (19).

### Visión estéreo

Los métodos de visión estereoscópica caen dentro de los denominados métodos pasivos, este tipo de métodos no interfieren en la escena que se está analizando. Los sistemas de visión estereoscópica artificial utilizan dos o más cámaras, en la mayoría de los métodos y procedimientos existentes en la literatura consultada se hace uso de modelos con dos cámaras.

El modelo de dos cámaras toma como referencia el propio modelo biológico de estereovisión, donde gracias a la distancia existente entre los dos ojos se puede establecer la distancia, esto es la tercera dimensión, a la que se encuentran los objetos.

Desde el punto de vista artificial, para obtener las imágenes desplazadas que permitan reconstruir la escena tridimensional existen dos aproximaciones básicamente:

- Utilizar dos o más cámaras todas ellas alineadas y separadas una cierta distancia. Esta distancia puede ser cualquiera, depende de la aplicación, pero debe ser conocida con exactitud.
- Utilizar únicamente una cámara móvil. Esta cámara debe ser capaz de desplazarse en línea recta y tomar imágenes mientras realiza este desplazamiento. Al igual que en el caso anterior, se debe conocer el desplazamiento de forma precisa que ha realizado la cámara entre la captura de dos imágenes consecutivas.

Otro aspecto inherente a los sistemas de visión estereoscópica es su geometría, pudiéndose optar por una geometría con los ejes ópticos paralelos o convergentes. El sistema visual humano trabaja fundamentalmente con ejes convergentes de forma que enfoca los ojos hacia los objetos de interés. Cuando el objeto está próximo se produce la convergencia de ejes sobre dicho objeto, mientras que si el objeto se sitúa en la lejanía prácticamente no existe convergencia de los ojos, pudiéndose en este caso decir que los ejes ópticos se sitúan en paralelo. Se puede utilizar el sistema de visión humano para comprobar algunas

características, ya que ha sido ampliamente estudiado, tanto en el campo de la biología como en el campo de la medicina. Así se comprobaría el desplazamiento que sufren dos imágenes de la misma escena al cambiar la perspectiva. Si con un ojo tapado y fijándonos en un objeto, nos destapamos dicho ojo y nos tapamos el que teníamos descubierto, podremos comprobar el desplazamiento relativo que sufre el objeto al ser observado mediante un ojo (se puede asimilar a una cámara) y mediante el otro. Este desplazamiento es menor cuanto más alejado esté el objeto de nosotros y será mayor cuanto más próximo se encuentre.

### Pasos en el proceso de la visión estereoscópica

Cuando se trabaja en visión estereoscópica se trabaja con la adquisición de imágenes y con la modelación de las cámaras. Una descomposición en mayor detalle del proceso de visión estereoscópica fue realizada por Barnard y Fischler en 1982. Según esta descomposición, el proceso completo de visión estereoscópica contempla seis pasos principales, a saber:

1. Adquisición de imágenes.
2. Modelado de la cámara (geometría del sistema).
3. Extracción de las características.
4. Correspondencia de las imágenes (características).
5. Determinación de la distancia (profundidad).
6. Interpolación, cuando sea necesaria.

Los pasos anteriores son de naturaleza secuencial con el orden en el que aparecen (Figura 19).

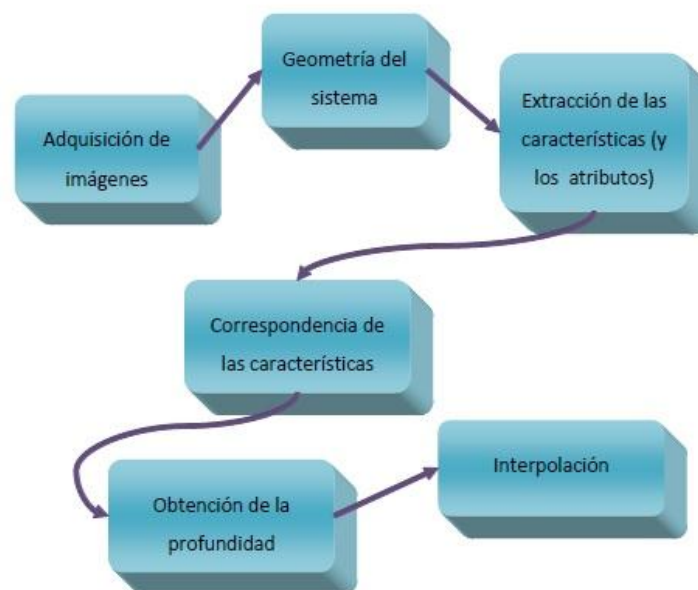


Figura 19. Principales pasos en la visión estereoscópica (19).

## 1.8 Calibración de un sistema de visión por computador

Uno de los aspectos importantes cuando se va a implementar un sistema de visión por computador utilizando cámara es la calibración, pues tiene gran influencia en la precisión y exactitud del *tracking*. La

calibración de cámara comprende dos aspectos distintos. En primer lugar es necesario determinar la relación que existe entre la posición respecto a la cámara de los puntos observados y la imagen que de ellos se obtiene.

Las técnicas de calibración de cámara se clasifican en (20):

- **Técnicas de optimización lineal.** Su principal ventaja es la simplicidad del modelo empleado, que revierte en un algoritmo de cómputo simple y rápido. Mediante un proceso de ajuste por mínimos cuadrados se determina la matriz que relaciona las coordenadas tridimensionales de los puntos de control y las de sus imágenes. En su contra tienen que no son aptos cuando se introduce la distorsión de la lente como un factor a calibrar, limitando por tanto la exactitud de las medidas realizadas, y la dificultad aparejada a la obtención de los parámetros a partir de la matriz calculada.
- **Técnicas de optimización no lineal.** Cuando el modelo empleado para la cámara se aleja del *pin-hole* básico y se introducen parámetros que reflejan la distorsión causada por la presencia de lentes.

Otras bibliografías como (21) aportan otras clasificaciones:

#### Computación lineal vs no lineal:

- **Lineal:** usan técnicas de resolución de sistemas de ecuaciones lineales, son muy simples de implementar y muy rápidos (como *el Direct Linear Transform*, algoritmo creado en 1971 para calibrar cámaras).
- **No lineal:** se basan en el uso de métodos iterativos, como el algoritmo *Gold Standard*. Generalmente se requiere una buena aproximación inicial obtenida de un método lineal. Son mucho más lentos, pero permiten resolver modelos de cámara más complejos que incluyen un mayor número de parámetros.

#### Calibración explícita vs implícita:

- **Explícita:** se obtienen los valores de cada uno de los parámetros que forman el modelo.
- **Implícita:** se obtienen generalmente matrices de transformación que contienen el conjunto de todos los parámetros. No se puede conocer el valor exacto de algunos parámetros.

La calibración de una cámara se divide en dos fases. En la primera de ellas se modela matemáticamente el comportamiento físico y óptico del sensor, definiendo un conjunto de parámetros intrínsecos y extrínsecos del modelo. En la segunda fase se estiman los valores de los parámetros del modelo obtenido en la fase anterior. Esta segunda fase se divide a su vez en el siguiente conjunto de tareas:

1. Situar un conjunto de referencias en la escena y capturar una serie de imágenes de la misma para determinar con precisión la posición de un conjunto de puntos tridimensionales.
2. Determinar la correspondencia entre los puntos proyectados en la imagen y la referencia correspondiente en la escena.
3. Ajustar los parámetros del modelo que resuelve la correspondencia entre puntos en la imagen y referencias en la escena.

### 1.8.1 Tipos de calibración de cámara

- **Calibración intrínseca de la cámara**, consiste en extraer de la matriz de calibración obtenida de una imagen, las características intrínsecas son: propiedades internas como la distancia focal, centro de proyección, no cambian si se mueve la cámara (21).
- **Calibración extrínseca de la cámara** consiste en extraer de la matriz de calibración obtenida de una imagen, las propiedades siguientes: vectores de traslación y rotación relativos al movimiento (21).

### 1.8.2 Características de los métodos de calibración

Un método de calibración debe cumplir las siguientes características (22):

- **Autonomía:** No debe requerir intervención por parte del usuario del sistema de visión por computador.
- **Precisión:** Debe lograr precisión requerida por el sistema.
- **Eficacia:** El procedimiento completo de calibración debe tener un coste computacional asumible por el sistema.
- **Versatilidad:** Debe poder calibrar el sistema aun con diferentes ópticas y distintos niveles de precisión.

### 1.8.3 Técnicas de calibración

Existen diversas técnicas de calibración de cámara. De acuerdo a la literatura estas pueden ser clasificadas en dos grandes categorías: calibración fotogramétrica y autocalibración (23):

- **Calibración fotogramétrica:** Se realiza mediante la observación de patrones cuya geometría en el espacio 3D es conocida con un buen nivel de precisión. Los patrones de calibración normalmente están posicionados en dos o tres planos ortogonales entre ellos. En algunos casos, basta con un único plano, cuya traslación es perfectamente conocida (24) y (23). Este tipo de calibración requiere una configuración elaborada, pero sus resultados son eficientes.
- **Autocalibración:** Este método se basa en el movimiento de la cámara observando una escena estática, a partir de sus desplazamientos y usando únicamente la información de la imagen. La rigidez de la escena impone en general restricciones sobre los parámetros de cámara. Tres imágenes tomadas por una misma cámara con parámetros intrínsecos fijos son suficientes para obtener tanto los parámetros extrínsecos como intrínsecos.

Dos métodos comunes de calibración, en la categoría de calibración fotogramétrica, son “Calibración directa de parámetros” y “Recuperación desde la homografía o matriz de proyección”. A través de la “Calibración directa de parámetros” se obtienen los parámetros extrínsecos e intrínsecos directamente, y a través de la “Recuperación desde la matriz de proyección” se obtienen los parámetros como una solución de “manera cerrada”, es decir, no se obtienen directamente los valores de los parámetros, sino que se calculan a partir de los valores de la homografía o matriz de proyección. Los métodos de calibración directa requieren de patrones de calibración consistentes, en al menos dos planos ortogonales. Sin embargo, la calibración mediante homografía o matriz de proyección, una variante recientemente propuesta, consiste en utilizar patrones que descansan en un único plano.

## 1.9 Bibliotecas para el desarrollo de la visión por computador

A continuación se realiza una descripción de bibliotecas que sirven como base para desarrollar aplicaciones que utilizan técnicas de visión por computadora.

### 1.9.1 OpenCV

*OpenCV (Open Source Computer Vision Library)* (25) es una biblioteca de tratamiento de imágenes, destinada principalmente a aplicaciones de visión por computadora en tiempo real. La biblioteca está escrita en C y C++, se ejecuta bajo *Linux*, *Windows* y *Mac OS X*. Se han desarrollado también “convertidores” para lenguajes como *C#*, *Python*, *Ruby* o *Java*, *Matlab* y otros lenguajes.

*OpenCV* se caracteriza por implementar una gran variedad de herramientas para la interpretación de la imagen, siendo la *IplImage* su principal estructura para el tratamiento de las mismas. Es compatible con *Intel Image Processing Library (IPL)* que implementa algunas operaciones en imágenes digitales y permite optimizar las funcionalidades en más de la mitad de las estructuras. *OpenCV* es principalmente una biblioteca que implementa algoritmos para: la calibración de cámara, la detección de rasgos, rastrear (Flujo Óptico), el análisis de la forma (Geometría, Contorno que Procesa), el análisis de movimientos (Plantillas del Movimiento, Estimadores), la reconstrucción 3D (Transformación de vistas), la segmentación de objetos y el reconocimiento (Histograma, etc.).

### 1.9.2 ArToolkit

Es una biblioteca para el desarrollo de aplicaciones de realidad aumentada en tiempo real. Es de libre uso para fines no comerciales y fue desarrollada por el Dr. Hirokazu Kato de la Universidad de Osaka en Japón. Principalmente incluye funciones de seguimiento y reconocimiento, aparte del *render OpenGL* para los objetos sintéticos. Es una biblioteca originalmente en C++, aunque permite convertir código a otros lenguajes como *Java* o *Python*.

*ArToolkit* utiliza técnicas de visión por computadora para calcular la posición y orientación de la cámara respecto a un patrón definido por la biblioteca, permitiendo al programador conocer donde se encuentra exactamente el mismo dentro de la escena.

Entre las funcionalidades que se pueden lograr con esta herramienta se encuentran:

- Seguimiento y orientación individual de la cámara.
- Utiliza códigos de seguimientos sencillos y optimizados.
- Detecta patrones cuadrados dentro de una imagen.
- Sencillo código de calibración de la cámara.

Las funcionalidades de *ArToolkit*, además de permitir crear aplicaciones de realidad aumentada, son muy eficientes para desarrollar sistemas de seguimiento basado en marcas.

### 1.9.3 Touchless

Se trata de una biblioteca para la plataforma .NET que permite crear aplicaciones *multi-touch* con una cámara web como interfaz. *Touchless* incluye una aplicación de prueba que se puede descargar y capturar gestos que luego serán reconocidos. Se caracteriza por:

- Disponible solo para.
- Reconocimiento de gestos humanos a través Microsoft Windows de cámara web.

*Touchless* es un experimento no concluido y en pleno desarrollo que aún no es nada sofisticado. Es un motor sencillo que explota técnicas no muy avanzadas de reconocimiento de imágenes para buscar objetos llamativos en las imágenes capturadas por la cámara web y usarlos como marcadores que luego se utilizarán para interactuar con la aplicación.

### 1.9.4 Comparación entre las bibliotecas

Luego de analizar las diferentes bibliotecas para el desarrollo de visión por computador se llegó a la conclusión que la biblioteca para el tratamiento de imágenes OpenCV es la que se utilizará para el desarrollo de la aplicación ya que es una biblioteca multiplataforma y además, contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica; siendo algunos de los aspectos anteriormente mencionados los que se utilizarán para el desarrollo de este trabajo.

## 1.10 Metodología de Desarrollo de Software

### XP (*Extreme Programming*)

En todo proceso de desarrollo de software se hace necesario el uso de una guía que posibilite el progreso eficiente en la construcción de aplicaciones. Como procedimientos y técnicas para toda la documentación y generación de artefactos del ciclo de vida del desarrollo del software, se requiere el uso de una metodología.

Una metodología para el desarrollo de un proceso de software es un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentos y aspectos de formación para los desarrolladores de software.

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

El desarrollo bajo XP tiene enumeradas características como son ([26](#)):

- Desarrollo iterativo e incremental.
- Pruebas unitarias continuas.

- Integración del equipo de programación con el usuario.
- Corrección de todos los errores.
- Refactorización del código.
- Propiedad del código compartida.

Entre las características más significativas de XP se encuentran (26):

- Orientado a la persona que produce y usa el software.
- Reduce el costo del cambio en las etapas de vida del sistema.

La metodología XP se basa en cuatro valores imprescindibles para el desarrollo de software (26):

- **Simplicidad:** enfocado en un diseño sencillo del código generado.
- **Comunicación:** potenciada por el desarrollo en pares, la presencia del cliente y la simplicidad en cuanto al código.
- **Retroalimentación:** propiciada por el protagonismo del cliente que participa activamente y por el trabajo en ciclos cortos.
- **Coraje:** enfrentando decisiones, en ocasiones complejas, que pudieran afectar el tiempo de desarrollo y la calidad del producto.

El ciclo de vida ideal consta de 6 fases (26):

- **Exploración:** los clientes plantean a grandes rasgos las Historias de usuario que son de interés para la primera entrega del producto.
- **Planificación de Entregas:** se establece la prioridad de cada Historia de usuario y los programadores realizan una estimación del esfuerzo necesario de cada una de ellas.
- **Iteraciones:** incluye varias iteraciones sobre el sistema antes de ser entregado. El plan de entrega está compuesto por iteraciones de no más de tres semanas.
- **Producción:** requiere de pruebas adicionales y revisiones del rendimiento antes de que el sistema sea trasladado al entorno del cliente.
- **Mantenimiento:** mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones.



## CAPÍTULO 2. Solución Propuesta

### Introducción

En este capítulo se describe la propuesta de solución para resolver el problema planteado en la investigación luego de haber realizado el estudio teórico. Se proponen las herramientas seleccionadas para el desarrollo de la aplicación, las técnicas y algoritmos básicos de detección de rostros, calibración de cámaras, obtención de mapas de disparidad y todo lo referente a la profundidad de los rostros detectados en la imagen.

### 2.1 Algoritmo de Viola y Jones

En el trabajo de Viola y Jones (27) se presenta un método supervisado, muy robusto y rápido para la detección de rostros. El mismo posee dos etapas principales: una de entrenamiento del detector, donde se construyen los clasificadores que lo componen y otra de detección, donde se emplea el detector entrenado en la primera etapa para detectar las caras en una imagen.

En el método de *AdaBoost* de Paul Viola y Michael Jones se presenta un nuevo concepto llamado la “Imagen Integral”, también presenta una extracción de valores llamados “Características *Haar*”, que no son más que valores extraídos al aplicar filtros con base *Haar* sobre la imagen integral, otro aporte es la combinación de clasificadores en cascada.

El sistema de detección de Viola-Jones utiliza grupos de características simples para llevar a cabo la detección. El uso de estas características posibilita una velocidad mucho mayor a la hora de detectar caras frente a un sistema basado en píxeles y beneficia la codificación en el dominio *ad-hoc*. Las características utilizadas son parecidas a aquellas del tipo *Haar* relacionadas con las propuestas por *Papageorgiou* en 1998.

Viola y Jones utilizan 4 tipos de características en su sistema de detección: dos tipos basados en dos rectángulos, un tipo basado en tres rectángulos y un tipo basado en cuatro rectángulos (Figura 20).

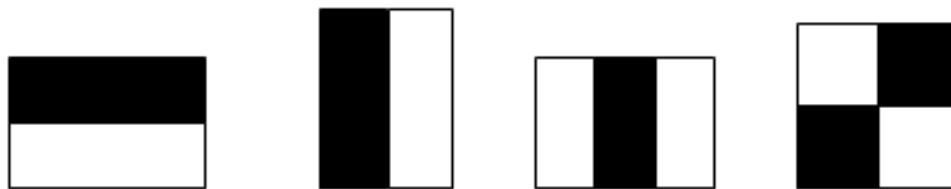


Figura 20. Características utilizadas en el detector de Viola-Jones (1).

De manera simplificada, las características pueden ser vistas como evaluaciones de la intensidad de conjuntos de píxeles. La suma de la luminancia de los píxeles en la región blanca se resta de la suma de los píxeles en la región oscura. El valor obtenido mediante la diferencia es el valor de la característica y puede combinarse con otros formando hipótesis en regiones de una imagen.



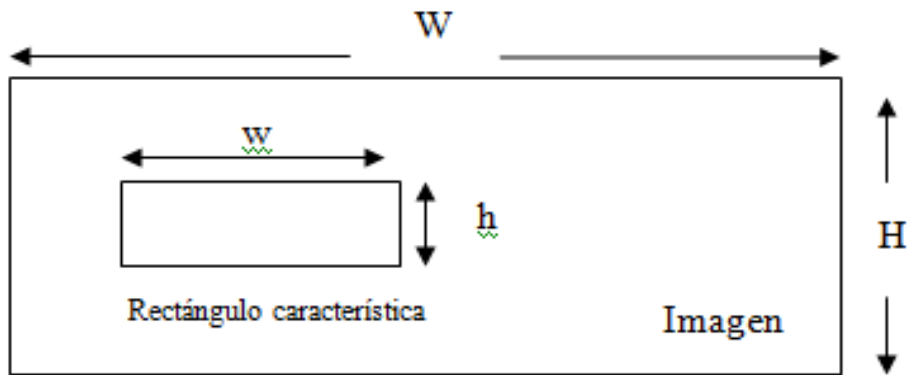


Figura 21. Representación de una imagen genérica conteniendo un modelo de característica (1).

Un hecho de particular relevancia y no muy desarrollado en las publicaciones de Viola y Jones, es la selección del conjunto de características a partir de las imágenes de entrenamiento, en su caso de 24x24 píxeles. El número de características de cada tipo para cada imagen es muy grande y de ellas se eligen las más adecuadas para la detección. Un método para calcular el número de características de cada tipo es el propuesto a continuación extraído de la implementación de Lienhart y Maydit (5). Para realizar estos cálculos, se hace uso de la Figura 21 que nos sirve como modelo para identificar los parámetros utilizados en la ecuación de Lienhart y Maydit:

$$F = X \cdot Y \left( W + 1 - w \frac{X + 1}{2} \right) \left( H + 1 - h \frac{Y + 1}{2} \right)$$

En la ecuación se puede identificar:  $\left[ X = \frac{W}{w} \right]$ ,  $\left[ Y = \frac{H}{h} \right]$ , siendo  $X$  e  $Y$  los valores máximos de escalado de la característica,  $W$  y  $H$  la anchura y altura máximas de la imagen respectivamente,  $w$  y  $h$  la anchura y altura máximas de la característica y  $F$  el número de características que se pueden extraer de dicha imagen. Aplicando la fórmula a las características utilizadas por Viola y Jones descritas en la Figura 20 y basándose en los datos resumidos se obtiene el número de características por cada tipo.

### 2.1.1 Imagen integral

A la hora de crear un sistema de detección facial resulta crucial encontrar un compromiso entre velocidad y eficiencia. Mediante el uso de una nueva representación de las imágenes, llamada imagen integral, Viola y Jones describen un método de evaluación de características de manera efectiva y a mayor velocidad. El concepto de la imagen integral es fácilmente comprensible. Esta estructura se construye tomando la suma de los valores de luminancia de los píxeles que se encuentran por encima y a la izquierda de un cierto punto en la imagen. Viola y Jones presentan la imagen integral como la integral doble de una imagen primero a lo largo de las filas y después a lo largo de las columnas. La imagen integral en el punto  $(x,y)$  viene dada por:

$$ii(x,y) = \sum_{a \leq x, b \leq y} i(a,b)$$

Donde  $ii(x,y)$  es la imagen integral y  $i(x,y)$  es el valor de la imagen en unas coordenadas específicas (Figura 22).

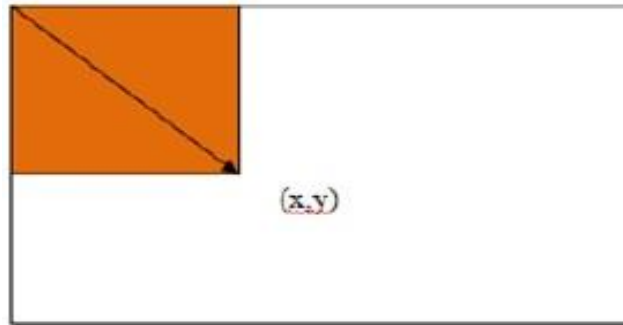


Figura 22. El valor de la imagen integral en un punto  $(x,y)$  es la suma de los píxeles de arriba a la izquierda (1).

La importancia de la imagen integral se basa en la capacidad de calcular rápidamente la suma de píxeles dentro de un área determinada de la imagen. Cualquier suma dentro de un área de la imagen puede calcularse utilizando cuatro referencias (Figura 23). Por lo tanto la diferencia entre dos regiones puede calcularse utilizando 8 referencias dentro de la imagen. Sin embargo, teniendo en cuenta que, por ejemplo los dos primeros tipos de características utilizan dos regiones rectangulares adyacentes la diferencia puede realizarse utilizando 6 referencias, en el caso del tercer tipo se utilizarían 8 referencias y en el cuarto tipo 9 referencias.

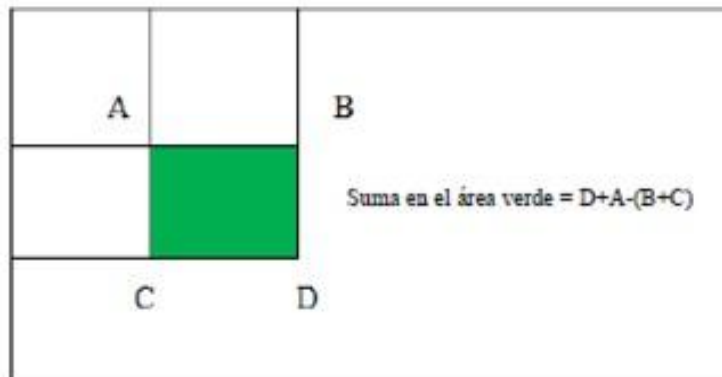


Figura 23. Cálculo del valor de la suma de los píxeles dentro de un área (1).

Una vez encontradas las características dentro de una imagen, más bien dentro de un conjunto de imágenes de entrenamiento, el objetivo del sistema es encontrar aquellas características que mejor definan una cara y nos ayuden a localizarla en una imagen. La hipótesis planteada en las publicaciones de Viola y Jones establece que un pequeño número de esas características pueden combinarse para formar un clasificador. Para ello establecen una variante del algoritmo *AdaBoost* de Freund y Shaphire (6). El algoritmo se utiliza para mejorar el rendimiento de un algoritmo de aprendizaje simple. Este algoritmo simple se llama clasificador simple o *weak learner*.

### 2.1.2 Clasificador simple

La base del sistema de Viola y Jones es la combinación de clasificadores simples o *weak learners* para obtener una clasificación eficiente de los datos. En relación a esto, un *weak learner*,  $h_j$ , es una estructura

simple que contiene una característica  $f$ , un umbral  $\theta$  y una paridad  $p$ . La salida del clasificador es binaria y depende de si el valor de una característica se encuentra por encima o debajo de un umbral.

Los clasificadores simples que se utilizan pueden verse como nodos de decisión en estructuras en árbol. Teniendo en cuenta que hay una gran cantidad de características en una imagen de 24x24 píxeles, el algoritmo *AdaBoost* debe seleccionar las características que mejor diferencien entre caras y no caras dentro de un conjunto de imágenes, y en él recalca la mayor parte del trabajo del entrenador.

La clasificación es un componente muy importante en los sistemas inteligentes. El problema que se plantea a la hora de realizar una clasificación consiste en decidir a qué clase pertenece un objeto. Sobre dicho objeto se realizaron varias mediciones, que son las llamadas características. Por ello, lo que interesa es aprender y encontrar una relación entre dichas características y las diversas clases a las que se enfrentan. Para implementar los clasificadores simples, en cada uno de los clasificadores fuertes o *strong classifiers* se tienen varias opciones utilizando diversos sistemas de aprendizaje: *SVM*, *perceptron*, *MLP*, *Naive Bayesian Classifiers*.

La selección de características se implementa del siguiente modo: Para cada ronda de *boosting*:

1. Evaluar cada función, aplicando cada filtro rectangular, sobre cada imagen de ejemplo.
2. Ordenar las imágenes según los valores obtenidos en el paso anterior.
3. Seleccionar el mejor umbral para cada característica.
4. Seleccionar el mejor filtro/umbral, es decir, la mejor característica.
5. Actualizar los pesos.

El método puede verse contextualizado dentro del algoritmo *AdaBoost* descrito en el siguiente sub-epígrafe.

### 2.1.3 Algoritmo AdaBoost

El algoritmo *AdaBoost* (Figura 24), se utiliza para seleccionar los mejores clasificadores simples. Teniendo en cuenta que existe un clasificador simple por cada característica se tiene que evaluar un total de  $K \cdot N$  clasificadores siendo  $K$  el número de características por imagen y  $N$  el número de imágenes que contenga el entrenamiento.

La ventaja principal de *AdaBoost* es su velocidad de aprendizaje. En este algoritmo los pesos se asignan de manera que favorezcan la clasificación de las caras, consiguiendo que estos ejemplos tengan mayores pesos o importancia.

Las mejores características se eligen basándose en el error ponderado que se produce. Este error ponderado es una función que utiliza los errores pertenecientes a los ejemplos de entrenamiento. El peso de un ejemplo clasificado correctamente se modifica mientras que el peso de un ejemplo mal clasificado se mantiene constante. Con esto se consigue que sea más difícil que la segunda característica clasifique por error un ejemplo que haya sido clasificado erróneamente por la primera característica frente a un ejemplo clasificado correctamente por esa primera característica. Otra manera de ver esto sería que la segunda característica, y las sucesivas, se ven forzadas a tomar más en cuenta los ejemplos clasificados erróneamente por características anteriores.

- Dadas imágenes de ejemplo  $(x_1, y_1), \dots, (x_n, y_n)$  donde  $y_i = -1; 1$  para ejemplos negativos o positivos de caras respectivamente, y  $T$  el número de hipótesis a construir.
- Inicializa los pesos  $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$  para  $y_i = -1; 1$  respectivamente, donde  $m$  y  $l$  son el número de ejemplos negativos y positivos respectivamente.
- For  $t = 1, \dots, T$

1. Normalizar los pesos,  $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$
2. Seleccionar el mejor clasificador simple respecto al error ponderado

$$\epsilon_t = \min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|$$

3. Definir  $h_t(x) = h(x, f_t, p_t, \theta_t)$ , donde  $f_t, p_t, \theta_t$  son los valores que minimizan  $\epsilon_t$ .
4. Actualizar los pesos:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

donde  $e_i = 0$ , si la imagen  $x_i$  esta correctamente clasificada, o  $e_i = 1$  en caso contrario; y  $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

- El clasificador final sería:

$$C(x) = \begin{cases} 1 & \text{si } \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{en caso contrario} \end{cases}$$

donde  $\alpha_t = \log \frac{1}{\beta_t}$

Figura 24. Algoritmo AdaBoost en pseudocódigo (1).

Después de sucesivas iteraciones del algoritmo el resultado es un conjunto de hipótesis ponderado que conforma un clasificador fuerte. La hipótesis final encontrada después de  $T$  iteraciones se obtiene al concluir el algoritmo, donde la clasificación binaria (cara contra no-cara) se realiza en función de los pesos individuales  $\alpha$  (representando los umbrales) y la suma del producto de cada peso y la clasificación de cada ejemplo en particular  $\alpha_t(x)$ .

### 2.1.4 Detector mediante clasificadores en cascada

El principio básico del algoritmo de detección facial de Viola y Jones consiste en escanear el detector muchas veces a través de una misma imagen, en diferentes posiciones y a distintas escalas. Incluso si una imagen contiene muchas caras está claro que la mayoría de las sub-ventanas que se escaneen no contendrán ninguna cara. Esto lleva a una nueva manera de ver el problema: “En vez de encontrar caras, el algoritmo debería descartar no-caras”.

La idea subyacente se basa en que es más fácil descartar una imagen que no contenga una cara que encontrar una cara en una imagen. Con esto en la cabeza, parece ineficiente construir un detector que contenga un solo clasificador fuerte ya que el tiempo de clasificación será constante sin importar la entrada del clasificador, ya que el clasificador tiene que evaluar todas las características que lo forman. Aumentar la velocidad de clasificación generalmente implica que el error de clasificación aumentará inevitablemente,

ya que para disminuir el tiempo de clasificación se debería disminuir el número de clasificadores simples que se utilizan. Para evitar esto Viola y Jones proponen un método para reducir el tiempo de clasificación manteniendo los requerimientos de rendimiento del clasificador (Figura 25).

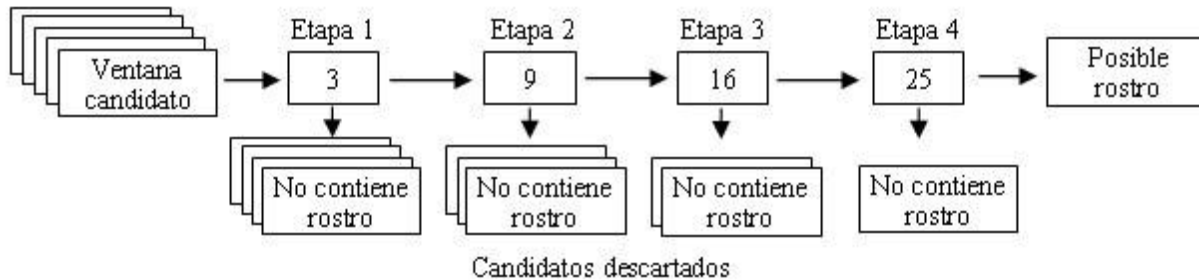


Figura 25. Estructura de clasificadores en cascada (1).

Este método consiste en el uso de una cascada de clasificadores fuertes. El trabajo en cada etapa del clasificador consiste en determinar si la sub-ventana que se analiza es definitivamente una “no-cara” o podría ser una cara. Cuando una sub-ventana es clasificada como no cara en alguna de las etapas del detector, se descarta inmediatamente. En caso contrario, si se clasifica como una posible cara, pasa a la siguiente etapa del clasificador. Se identificará una sub-ventana como contenedora de una cara si y sólo si pasa a través de todas las etapas del detector de forma positiva.

Si se utilizase un clasificador que tuviese un único estado, normalmente habría que aceptar los falsos negativos para reducir la tasa de falsos positivos. Sin embargo, para las primeras etapas del clasificador en cascada se acepta una alta tasa de falsos positivos esperando que las etapas posteriores puedan encargarse de reducir esta tasa mediante clasificadores más especializados. Con esto se pretende también reducir la tasa de falsos negativos en el clasificador final, ya que una sub-ventana será clasificada como cara sólo en el caso de que haya pasado por todas las etapas del clasificador.

Este método está implementado en la OpenCV a través de la función *cvHaarDetectObjects*, la cual entre sus parámetros recibe un clasificador en cascada haciendo uso de la funcionalidad *CvHaarClassifierCascade*, estos métodos son los que serán utilizados para detectar las caras en el entorno.

## 2.2 Información estéreo

La información estéreo es una de las técnicas más usadas en la actualidad ya que ofrece muy buenos resultados en el área de visión por computador, esta técnica permite determinar la distancia a la cámara de objetos y personas, requieren de al menos dos cámaras o un sistema especializado con este fin.

Para obtener una imagen de profundidad a partir de la utilización de la estereopsis se necesita la calibración de las cámaras a utilizar, con esta imagen de profundidad se logrará determinar cuán cerca o lejos está cada rostro detectado.

### 2.2.1 Calibración de cámara

Para obtener los parámetros de calibración será necesario obtener unas 20 fotografías a un patrón de calibración como se muestra en la Figura 26, donde el patrón deberá aparecer perfectamente enfocado de

forma que sean visibles todas las esquinas de la cuadrícula y ajustados para que el patrón abarque la mayor parte de la imagen posible, siempre dejando espacio para el reborde blanco del patrón.

El patrón debe fotografiarse de forma que la perspectiva de las cuadrículas fugue en todas la direcciones posibles (hacia la izquierda abajo, izquierda arriba, derecha arriba, de abajo a arriba, etc.). Todo esto es importante porque se utilizarán algoritmos de detección de esquinas y cuanto más información tenga, menor error cometerá en la calibración. Es importante también que todas las fotografías tengan condiciones lumínicas lo más parecidas posibles.

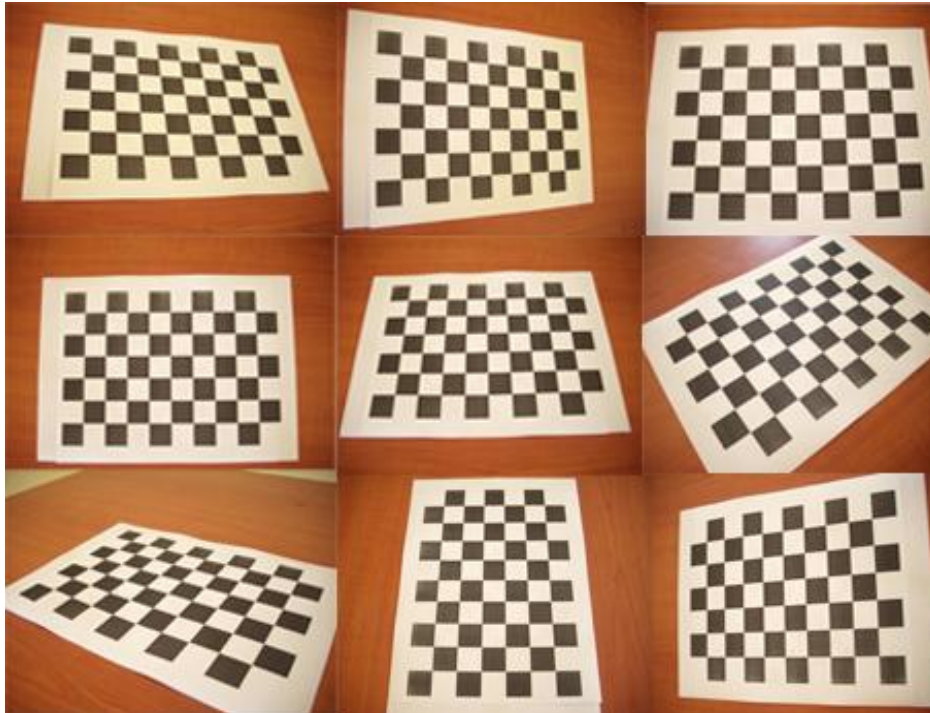


Figura 26. Ejemplos de patrones de calibración (19).

### Colocación de las cámaras

Una vez colocado todo el sistema y realizado las fotografías, se tendrá en cuenta la colocación de las cámaras en una posición que venga bien y teniendo en cuenta que jamás se modificará el zoom de las mismas, ya que se estaría anulando la calibración que se habrá realizado. Las cámaras se colocan en un trípode y a la hora de realizar las fotografías se trata de evitar cualquier contacto físico para asegurar que las cámaras estén bien estables.

### Movimiento de la cámara

Cuando se realice las dos fotografías se intentará que transcurra el mínimo tiempo posible entre ambas tomas. Se deberá tener en cuenta que para que la restitución tenga éxito el solapamiento de las imágenes debe ser por lo menos de un 70-80%. El movimiento de las cámaras se llevará a cabo de forma que los ejes de las cámaras estén paralelos entre una posición y la otra, apareciendo solo una disparidad horizontal entre ambas imágenes. Una forma de asegurarse es colocar marcas, para mover el trípode. A pesar de las precauciones que se deberán tomar, es inevitable cometer pequeños errores en la posición de las cámaras.



## 2.2.2 Selección del algoritmo de apareamiento estereoscópico

Mediante un algoritmo de apareamiento estereoscópico se determina el conjunto de correspondencias correctas entre al menos dos imágenes estereoscópicas con el fin de encontrar la profundidad de la escena en base a las diferencias de posición entre un conjunto de correspondencias.

La geometría epipolar del sistema estereoscópico limita la búsqueda de correspondencia a solo una dimensión. Mediante la restricción epipolar la correspondencia para un píxel en una imagen se encontrará sobre la línea epipolar de la otra imagen, esta búsqueda se simplifica aún más si se rectifican las imágenes (Figura 27), transformando así la línea epipolar en una línea de exploración horizontal común a las dos imágenes.

La comparación para cada píxel de una imagen se realiza utilizando todos los píxeles alrededor de este que estén dentro de una ventana de correlación mediante alguna función de similitud como la suma de diferencias al cuadrado (SSD), la correlación cruzada (NCC) o la suma de las diferencias absolutas (SAD). La búsqueda de correspondencia se realiza sobre la línea de exploración dentro de un rango de disparidad establecido, como las imágenes se encuentran rectificadas esta es una línea horizontal común a las dos imágenes. El mejor valor obtenido por la función de similitud determina a qué distancia o disparidad horizontal se encuentra el píxel analizado, respecto a la posición de su correspondiente en la otra imagen. Con cada apareamiento se crea un mapa de disparidad el cual indica para cada píxel la diferencia de posición a la que se encuentra su correspondencia en la otra imagen. En cada imagen existirán detalles que no se encuentran en la otra pues se encuentran ocluidos.

Para encontrar la profundidad real de la escena es necesario conocer las propiedades geométricas del sistema y así transformar las disparidades en unidades de longitud.



Figura 27. Par de imágenes estereoscópicas rectificadas.

## 2.2.3 Cálculo de disparidad

Los sistemas de visión estereoscópica permiten obtener distancias de objetos dentro de una escena a partir de un par de imágenes de la misma. Las distancias se obtienen a partir de la medición de la disparidad de

dos elementos semejantes en ambas imágenes. Las disparidades de todos los puntos de una imagen conforman el denominado mapa de disparidad como se muestran en la Figura 28 y Figura 29.

Si el par de imágenes estéreo ha sido previamente rectificado, las líneas epipolares conjugadas se vuelven colineales y paralelas a la línea base del sistema estéreo. Esto significa que puntos que se corresponden en ambas imágenes tienen la misma coordenada en el eje vertical y reduciendo el problema de correspondencia a la búsqueda en una sola dimensión (eje horizontal  $x$ ). La principal guía para resolver el “apareamiento” de un par de imágenes estéreo (problema de correspondencia), se basa en la semejanza o la no semejanza. Las características o áreas correspondientes deben ser similares en ambas imágenes del par estéreo.

Existen diferentes medidas de semejanza, difieren entre sí en el costo computacional, la eficiencia, entre otras. Básicamente se pueden dividir las diversas técnicas en dos grupos: las basadas en características, y las basadas en la correlación. Los métodos basados en características restringen la búsqueda a un conjunto disperso de características. En los métodos basados en la correlación, los elementos a comparar son ventanas de la imagen de dimensión fija y el criterio de semejanza es una medida de la correspondencia entre las ventanas de las dos imágenes. El elemento correspondiente queda determinado por la ventana que maximiza el criterio de semejanza dentro de la región de búsqueda.

Dentro de este segundo grupo se pueden distinguir tres métodos para el cálculo del mapa de disparidad: la suma de diferencias absolutas (*Sum of Absolute Differences: SAD*), la suma de diferencias al cuadrado (*Sum of Squared Differences: SSD*), y la correlación cruzada normalizada (*Normalized Cross Correlation: NCC*) (28). A partir del análisis de pares estereoscópicos de diferentes bibliografías, se observó que no existen grandes diferencias respecto a la calidad de los mapas de disparidades obtenidos por los diferentes métodos evaluados.

Para realizar el procesamiento del mapa de disparidad y obtener una alineación, se buscan puntos de correspondencia en regiones lejanas de ambas imágenes. En dichas regiones la disparidad es cero, por lo tanto las imágenes deben tener iguales coordenadas para dichos puntos.

## 2.2.4 Obtención del mapa de disparidad denso mediante la OpenCV.

Básicamente OpenCV proporciona 2 métodos para calcular un mapa de disparidad denso:

- ***cvFindStereoCorrespondenceBM***: Este método es rápido, puede procesar varias imágenes por segundo, pero si los parámetros no son ajustados, los resultados son pobres e imprecisos (Figura 28).
- ***cvFindStereoCorrespondenceGC***: Este método es muy lento, ya que tarda varios segundos, incluso minutos por imagen, pero se obtienen resultados muy precisos (Figura 29).



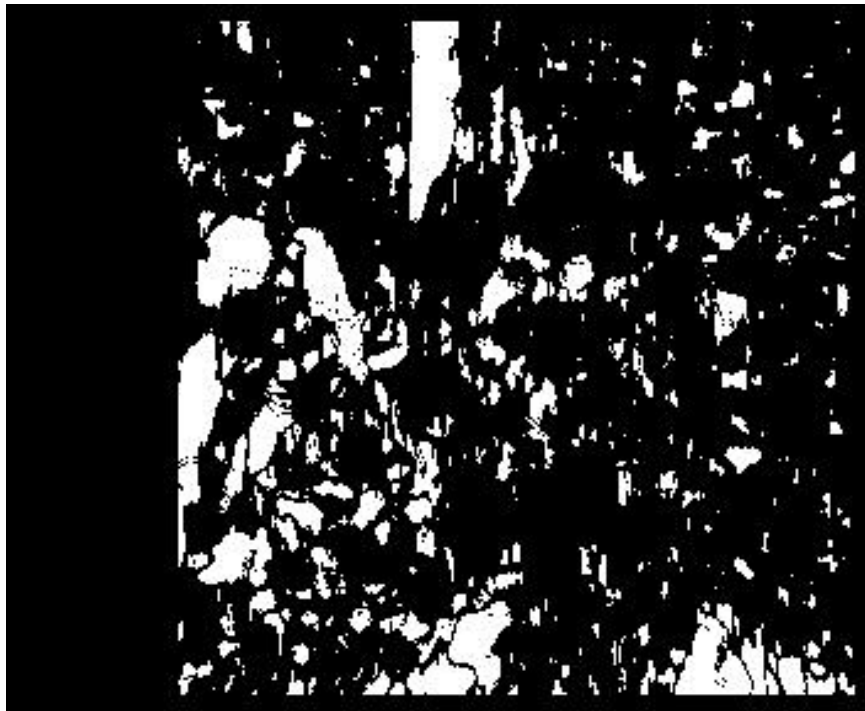


Figura 28. Imagen obtenida aplicando el método *cvFindStereoCorrespondenceBM*.

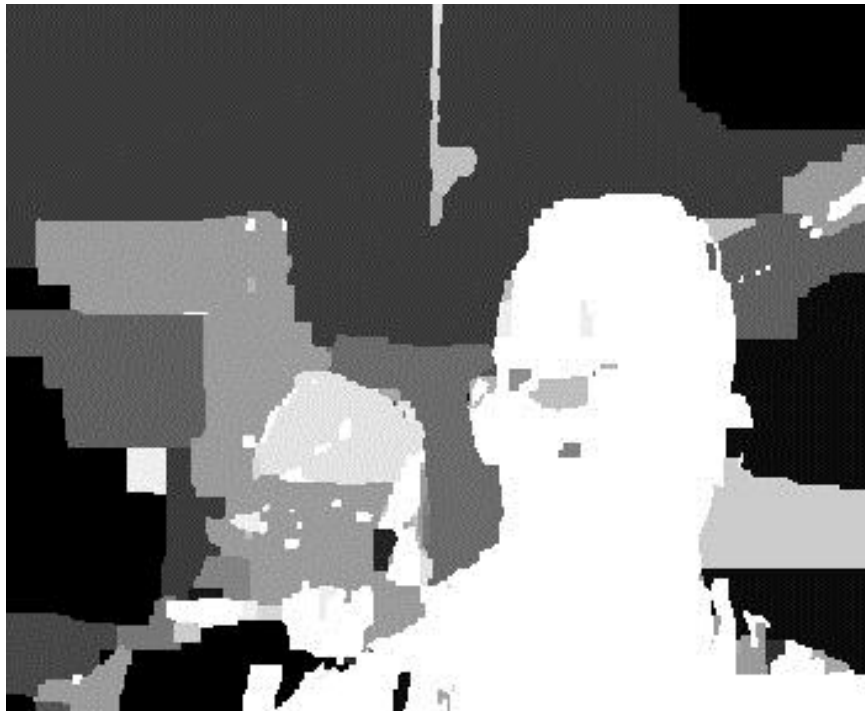


Figura 29. Imagen obtenida aplicando el método *cvFindStereoCorrespondenceGC*.

Luego de obtenerse los diferentes mapas de disparidad basándose en los métodos anteriormente mencionados se estableció que para el trabajo de cálculo de las posiciones relativas de las personas en el entorno es conveniente utilizar el *cvFindStereoCorrespondenceGC*, pues a pesar de que el mismo es lento en tiempo de ejecución aporta más información que el *cvFindStereoCorrespondenceBM*.

## 2.3 Obtención de profundidad de los rostros en la imagen original

Basándose en la información que brinda el mapa de disparidad (Figura 29) y la detección de rostros que se muestra en la Figura 30 luego de aplicar el método Viola y Jones, se logró identificar la posición relativa de los rostros en la imagen, como se muestra en la Figura 31, mostrando siempre con un valor menor el rostro más cercano a la cámara.

La obtención de la profundidad en la imagen se logró analizando la información que brinda el mapa de disparidad obtenido a través del cálculo del promedio de los píxeles en cada región en la cual se obtuvo un rostro en el proceso de detección, ya que los rostros mientras más cercanos a la cámara se encuentran, más clara sale la región que estos ocupan en el mapa de disparidad.

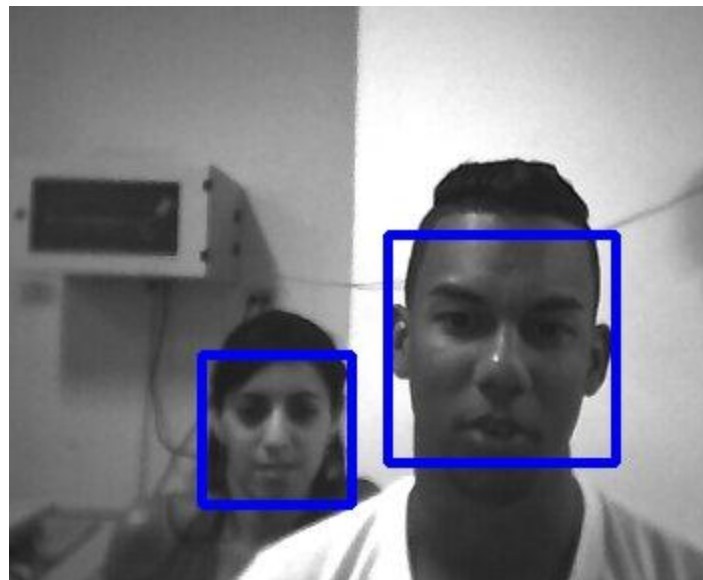


Figura 30. Imagen con detección de rostros



Figura 31. Posiciones relativas de los rostros en la imagen.

## 2.4 Obtención del mapa de oclusión

Con la determinación de la profundidad de las personas se requiere la representación de dicha información de tal forma que pueda ser de utilidad en posteriores análisis. Con este objetivo se utiliza una variante de mapa de oclusión (Figura 32), la cual, mediante una matriz binaria, representa los objetos, en este caso personas, que se encuentran en la escena.

La propuesta es representar no solo los puntos ocupados por las personas, sino incluir además la posición relativa de las mismas con respecto a la cámara.

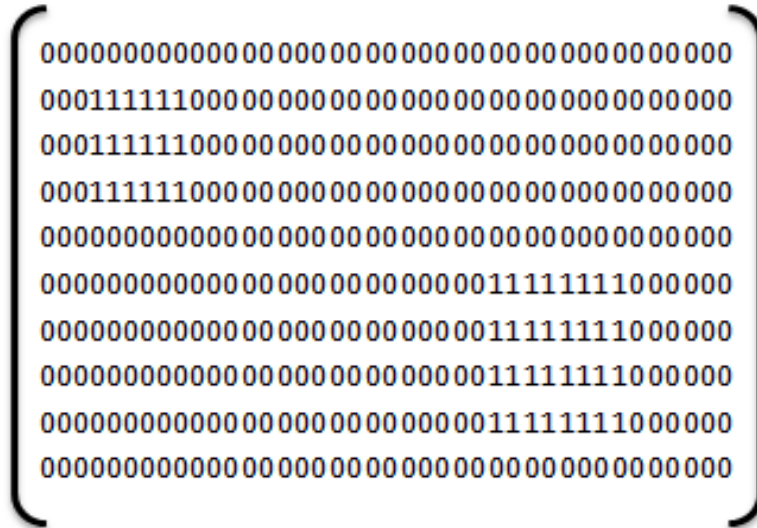


Figura 32. Imagen binaria

Para obtener el mapa de oclusión con la información de profundidad de los rostros detectados en la escena se crea una matriz con las mismas dimensiones de las imágenes de entrada, donde inicialmente cada píxel de la imagen en la matriz se encontrará en 0.

Luego, con la información de las coordenadas donde se encuentran los rostros, se procede a actualizar la matriz, colocando en cada región los números ascendentemente, comenzando siempre por 1 y representando con el mismo, a la persona que más cerca se encuentra de las cámaras (Figura 33). Se utiliza para ello la información que se obtuvo al calcular las posiciones relativas (Figura 31). La información que se obtiene al representar la profundidad de las personas se guarda en un fichero de texto.



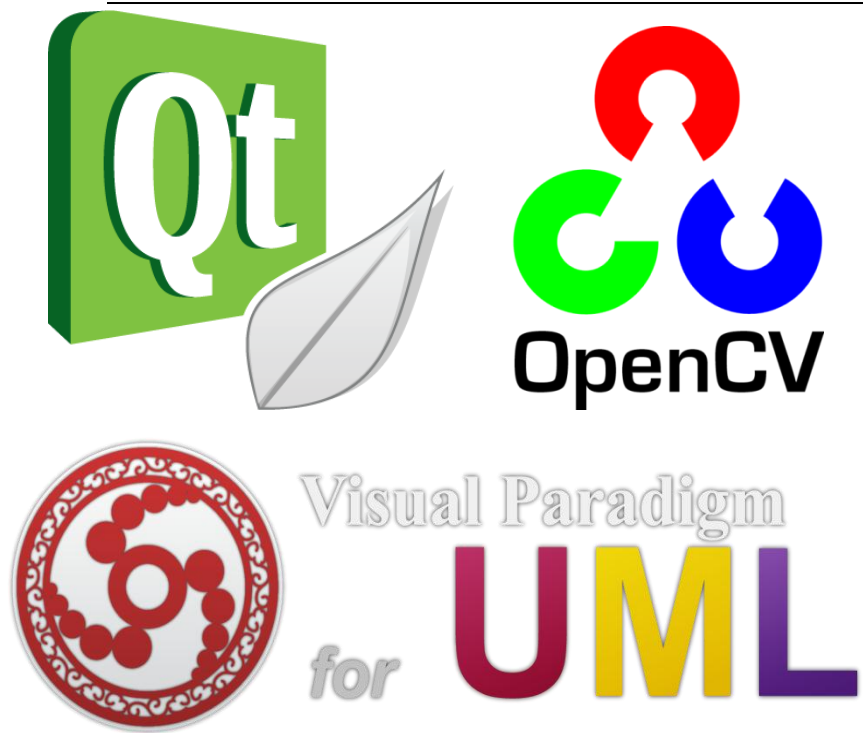


Figura 34. Herramientas utilizadas en el proceso de desarrollo

## CAPÍTULO 3. Ingeniería del sistema

### Introducción

En este capítulo se hará una descripción a nivel conceptual de la solución propuesta. Se definen las condiciones a tener en cuenta para el correcto funcionamiento del sistema, así como las historias de usuarios y demás artefactos que genera la metodología XP. Se tratarán los conceptos más importantes del área de interés mediante un modelo de dominio. Se describirán las capacidades que deberá cumplir el sistema a partir de los requisitos funcionales, así como las características del mismo expresadas por los requisitos no funcionales.

### 3.1 Modelo de dominio

La metodología XP se inspira en la simplicidad, basada en desarrollar sólo el sistema que realmente se necesita, sin embargo su gran adaptabilidad permite el empleo de diagramas UML, siempre y cuando influyan en el mejoramiento de la comunicación. Debido a ello se decide realizar un modelo de dominio (Figura 35) pues permite a los usuarios, clientes, desarrolladores e interesados, a utilizar un vocabulario común para lograr un efectivo entendimiento del contexto en que se encuentra el sistema proporcionando una perspectiva conceptual de los objetos implícitos en él. Se realiza su descripción a través de un diagrama de clases UML.

Según Ivar Jacobson, Grady Booch, y James Rumbaugh, “un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema” (29). Es una representación visual de clases conceptuales o de objetos reales en un dominio de interés.

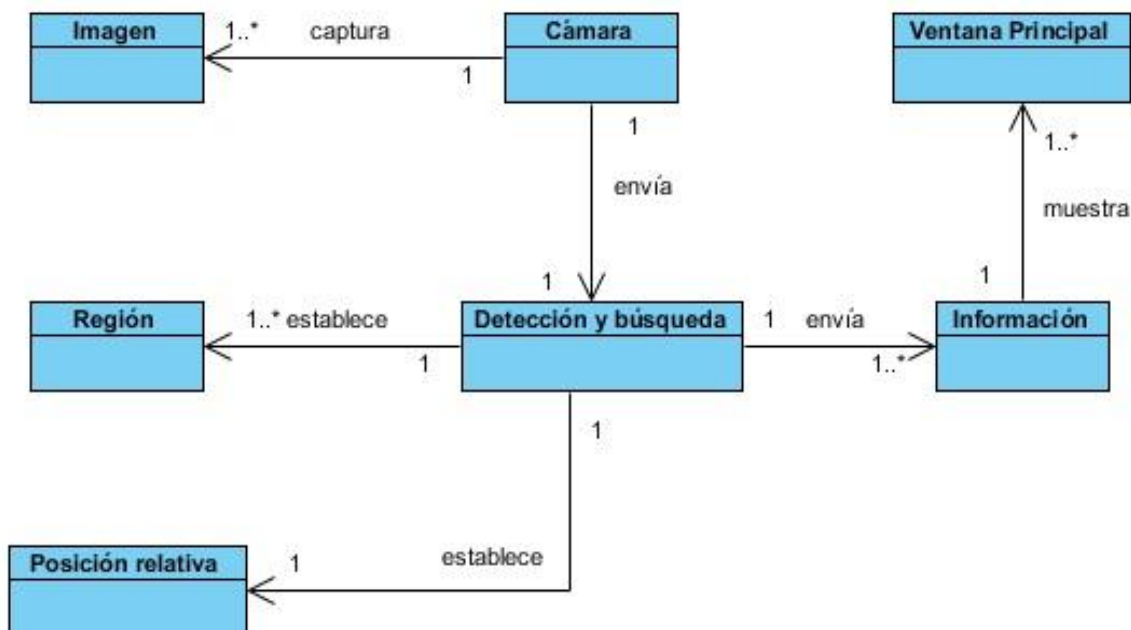


Figura 35. Modelo de Dominio.

### Definición de las clases del modelo de dominio.

- **Cámara:** Encargada de capturar las imágenes a través de la información estéreo.
- **Imagen:** Guarda todas las imágenes capturadas por la cámara web.
- **Detección y búsqueda:** Se encarga de tomar las imágenes y realizar todo el proceso de detección de rostros.
- **Región:** Pinta la región en las cuales se encuentran los rostros detectados.
- **Posición relativa:** Calcula la profundidad a la cual se encuentra cada rostro detectado con respecto a las cámaras.
- **Información:** Gestiona toda la información que será mostrada.
- **Ventana Principal:** Muestra toda la información recibida.

### 3.2 Personal relacionado con el Sistema

Una persona relacionada con el sistema no es más que aquella que está de una forma u otra vinculada al proceso de desarrollo de la aplicación e interactuando con el mismo.

Personal relacionado con el Sistema	Justificación
Desarrollador	Persona encargada de implementar e integrar las funcionalidades de la aplicación.
Usuario	Persona que va a interactuar con la aplicación.

Tabla 1. Personal relacionado con la aplicación.

### 3.3 Exploración. Historia de Usuario

Las Historias de Usuario (HU) son la técnica utilizada en XP para especificar y administrar los requisitos del software de una forma eficaz. Son descripciones cortas y escritas en el lenguaje del usuario sin terminología técnica, debido a ello son lo suficientemente comprensibles y delimitadas para que los programadores puedan implementarlas en unas semanas. Permiten responder rápidamente a los requisitos cambiantes pues su tratamiento es dinámico y flexible.

Para el presente trabajo se obtienen un total de 4 HU que son implementadas en una sola iteración. A continuación se muestran estas HU.

Historia de Usuario	
<b>No. 1</b>	<b>Nombre:</b> Detectar rostros en la imagen
<b>Usuario:</b> usuario	
<b>Programador responsable:</b> Andiel Muñoz Viamonte	

<b>Prioridad en el dominio:</b> Alta	<b>Nivel de Complejidad:</b> Alta
<b>Tiempo de Estimación:</b> 3 semanas	<b>Iteración Asignada:</b> 1
<b>Descripción:</b> En este proceso mediante el método de detección se busca detectar los rostros en la imagen.	
<b>Información adicional (Observaciones):</b> La aplicación localiza los rostros y pinta los ROI (Región de interés).	

Tabla 2. HU Detectar rostros dentro de la imagen.

Historia de Usuario	
No. 2	<b>Nombre:</b> Detectar rostros en video.
<b>Usuario:</b> usuario	
<b>Programador responsable:</b> Andiel Muñoz Viamonte	
Prioridad en el dominio: Alta	<b>Nivel de Complejidad:</b> Alta
Tiempo de Estimación: 3 semanas	<b>Iteración Asignada:</b> 1
<b>Descripción:</b> En este proceso mediante el método de detección se busca detectar los rostros en la imagen en tiempo de ejecución capturando los <i>frames</i> del video.	
<b>Información adicional (Observaciones):</b> La aplicación localiza los rostros en tiempo de ejecución y pinta los ROI.	

Tabla 3. HU Detectar rostros en video.

Historia de Usuario	
No. 3	<b>Nombre:</b> Obtener mapa de disparidad
<b>Usuario:</b> usuario	
<b>Programador responsable:</b> Lislien González Rodríguez	
Prioridad en el dominio: Alta	<b>Nivel de Complejidad:</b> Alta
Tiempo de Estimación: 0.6 semanas	<b>Iteración Asignada:</b> 1
<b>Descripción:</b> En este proceso se crean los mapas de disparidad a través de los dos métodos existentes (BM y GC).	



**Información adicional (Observaciones):** A partir de dos imágenes se crean los mapas de disparidad.

Tabla 4. HU Obtener mapa de disparidad.

Historia de Usuario	
<b>No. 4</b>	<b>Nombre:</b> Calcular posiciones relativas de los rostros detectados.
<b>Usuario:</b> usuario	
<b>Programador responsable:</b> Lislien González Rodríguez.	
<b>Prioridad en el dominio:</b> Alta.	<b>Nivel de Complejidad:</b> Alta.
<b>Tiempo de Estimación:</b> 2 semanas.	<b>Iteración Asignada:</b> 1
<b>Descripción:</b> Luego de que se detecten los rostros en la imagen se realiza el cálculo de posiciones relativas para determinar la profundidad de la imagen.	
<b>Información adicional (Observaciones):</b> La aplicación necesita reconocer si se detectó algún rostro en la imagen para poder realizar el cálculo de posiciones relativas en la misma. Utilizando para ello el mapa de disparidad.	

Tabla 5. HU Calcular posiciones relativas de los rostros detectados.

Las HU describen las funcionalidades que debe realizar la aplicación. Proveen información acerca de la prioridad de la funcionalidad a implementar así como el tiempo estimado de duración de dicha implementación. A continuación las HU son relacionadas en el artefacto Lista de Reserva del Producto en la cual se referencia información relevante acerca de las mismas.

### 3.4 Lista de reserva del producto

La lista de reserva del producto es una tabla que contiene los requisitos funcionales que debe cumplir la herramienta que se desea realizar, ordenados según la prioridad en el negocio (Alta, Media y Baja). Se indica de cada uno de ellos la estimación, su implementación por semanas y el rol que lo estimó. Contiene por último los requisitos no funcionales que requiere el sistema a desarrollar.

Elementos	Descripción	Estimación	Estimado por
<b>Prioridad: Alta</b>			
1.	Detectar rostros en la imagen	3 semanas	Desarrollador
2.	Detectar rostros en video.	3 semanas	Desarrollador

3.	Obtener mapa de disparidad	0.6 semanas	Desarrollador
4.	Calcular posiciones relativas de los rostros detectados.	2 semanas	Desarrollador
<b>Prioridad: Media</b>			
<b>Prioridad: Baja</b>			
<b>Requisitos No Funcionales:</b>			
1.	<b>Apariencia o interfaz externa:</b> Interfaz intuitiva, organizada, con una navegabilidad flexible y de fácil comprensión.		Desarrollador
2.	<b>Hardware:</b> Se necesita 1 GB de memoria RAM mínimo, un CPU Pentium 4 y dos cámaras.		Desarrollador
3.	<b>Rendimiento:</b> El tiempo de respuesta dependerá de la imagen a tratar. La aplicación debe tener un alto nivel de procesamiento y un buen manejo de la memoria de la máquina.		Desarrollador
4.	<b>Restricciones del diseño y la implementación:</b> Para el diseño e implementación de la aplicación se utiliza la metodología XP, haciendo uso del Visual Paradigm para el modelado. Se utiliza como lenguaje de programación C++ y como IDE de desarrollo el QtCreator.		Desarrollador

Tabla 6. Lista de reserva del producto.

### 3.5 Tareas de Ingeniería

Después de haberse realizado la definición de las HU, el equipo de desarrollo divide por cada una de ellas una serie de tareas de ingenierías que contribuyen al desarrollo de las HU. A continuación se presentan las tareas de ingenierías correspondientes a cada una de las HU.

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Importar	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.4 semanas
<b>Fecha Inicio:</b> 26/02/2013	<b>Fecha Fin:</b> 28/02/2013
<b>Programador Responsable:</b> Andiel Muñoz Viamonte	
<b>Descripción:</b> Para lograr el trabajo con imágenes se hace necesario importarlas para luego proceder a detectar los rostros encontrados en la misma, haciendo uso del método Importar() de la clase form_imagen.	

Tabla 7. Tarea de Ingeniería Num.1 correspondiente a la HU Detectar rostros en la imagen.

Tarea de Ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Convertir Imagen	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6 semanas
<b>Fecha Inicio:</b> 28/02/2013	<b>Fecha Fin:</b> 03/03/2013
<b>Programador Responsable:</b> Andiel Muñoz Viamonte	
<b>Descripción:</b> Para mostrar la imagen de formato <i>Iplimage</i> con la cual se trabaja en la OpenCV se hizo necesario utilizar el método ConvertirImagen(), el cual devuelve la imagen como un <i>Qimage</i> , para que pueda ser mostrada utilizando un componente de QtCreator.	

Tabla 8. Tarea de Ingeniería Num.2 correspondiente a la HU Detectar rostros en la imagen.

Tarea de Ingeniería	
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> 1
<b>Nombre Tarea:</b> Detección de Rostros	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 2 semanas
<b>Fecha Inicio:</b> 04/03/2013	<b>Fecha Fin:</b> 14/03/2013
<b>Programador Responsable:</b> Andiel Muñoz Viamonte	
<b>Descripción:</b> Se procede a detectar los rostros en la imagen haciendo uso del método DeteccionDeRostros().	

Tabla 9. Tarea de Ingeniería Num.3 correspondiente a la HU Detectar rostros en la imagen.

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 2
<b>Nombre Tarea:</b> Detección de Rostros en video	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 3 semanas
<b>Fecha Inicio:</b> 15/03/2013	<b>Fecha Fin:</b> 05/04/2013
<b>Programador Responsable:</b> Andiel Muñoz Viamonte	
<b>Descripción:</b> Se procede a detectar los rostros en video haciendo uso del método DeteccionCamara(). La aplicación localiza los rostros en tiempo de ejecución y pinta los ROI.	

Tabla 10. Tarea de Ingeniería Num.1 correspondiente a la HU Detectar rostros en video.

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 3
<b>Nombre Tarea:</b> Obtener mapa de disparidad	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6 semanas
<b>Fecha Inicio:</b> 25/04/2013	<b>Fecha Fin:</b> 28/04/2013

<b>Programador Responsable:</b> Lislien González Rodríguez
<b>Descripción:</b> Se crean los mapas de disparidad de dos formas (BM y GC) a través del método MapaDisparidad(). Siendo el GC el utilizado, pues aporta la información necesaria para el posterior trabajo.

Tabla 11. Tarea de Ingeniería Num.1 correspondiente a la HU Obtener mapa de disparidad.

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> 4
<b>Nombre Tarea:</b> Calcular promedio de píxeles en la imagen	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1 semana
<b>Fecha Inicio:</b> 29/04/2013	<b>Fecha Fin:</b> 06/05/2013
<b>Programador Responsable:</b> Lislien González Rodríguez	
<b>Descripción:</b> Se procede a calcular el promedio de píxeles de la región de detección haciendo uso del método PíxelesImagen().	

Tabla 12. Tarea de Ingeniería Num.1 correspondiente a la HU Calcular posiciones relativas de los rostros detectados.

Tarea de Ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> 4
<b>Nombre Tarea:</b> Ordenar lista de personas y pintar ROI.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.6 semana
<b>Fecha Inicio:</b> 29/04/2013	<b>Fecha Fin:</b> 31/04/2013
<b>Programador Responsable:</b> Lislien González Rodríguez	
<b>Descripción:</b> Se procede a ordenar la lista de personas tomando en cuenta el promedio de píxeles y luego se pinta la región de interés con la información de profundidad, haciendo uso del método OrdenarYPintar().	

Tabla 13. Tarea de Ingeniería Num.2 correspondiente a la HU Calcular posiciones relativas de los rostros detectados.

Tarea de Ingeniería	
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> 4
<b>Nombre Tarea:</b> Guardar mapa de oclusión.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.4 semana
<b>Fecha Inicio:</b> 1/05/2013	<b>Fecha Fin:</b> 03/05/2013
<b>Programador Responsable:</b> Lislien González Rodríguez	
<b>Descripción:</b> Se procede a guardar el mapa de oclusión en un archivo de texto, en el cual se representa las posiciones relativas de las personas con respecto a las cámaras, haciendo uso del método GuardarMapaOclusion().	

Tabla 14. Tarea de Ingeniería Num.3 correspondiente a la HU Calcular posiciones relativas de los rostros detectados.

### 3.6 Plan de Iteraciones

Las HU después de ser descritas, identificadas y estimado el esfuerzo propuesto para la realización de cada una de ellas, se especifica cuáles serán implementadas en cada iteración del sistema por lo que se procede a la realización de un plan de iteraciones.

La metodología XP aporta mayor valor a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. El plan de iteraciones empleado para la planificación, donde el cliente establece la prioridad de cada HU y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas.

Iteración	Descripción de la iteración	Orden de la HU a implementar	Duración total
1.	Tiene como objetivo desarrollar las HU con prioridad alta las cuales forman la base de la aplicación a desarrollar. Son las encargadas de realizar el proceso de realizar todo el proceso de detección facial tanto en imágenes como en video, obtener el mapa de disparidad y finalmente realizar el cálculo de las posiciones relativas en los rostros detectados.	HU 1  HU 2  HU 3  HU 4	8.6 semanas

Tabla 15. Plan de Iteraciones.

### 3.7 Modelo de Diseño

En la fase de diseño, XP propone mejorar la comunicación entre todos los integrantes del equipo, al crear una visión global y común de lo que se quiere desarrollar para lograr un diseño sencillo pues Kent Beck dice que *“en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos”*, (30) además Pressman asegura: *“el diseño es el lugar donde se fomentará la calidad del software”* (31).

Es preciso describir qué clases existen y cómo interactúan, para ello la metodología XP utiliza ciertas técnicas, llamadas tarjetas Clase, Responsabilidad y Colaboración (CRC). Sin embargo se puede hacer uso de diagramas UML, siempre y cuando influyan en el mejoramiento de la comunicación y se enfoquen en la información elemental. Por dicha razón se realiza un diagrama de las clases a implementar para mostrar sus relaciones y dependencias utilizando notación UML.

#### 3.7.1 Diagrama de Clases

El diagrama de clases (Figura 36) facilita una abstracción de la implementación del sistema y admite una mejor comprensión de los atributos, relaciones y métodos que contienen las clases, obteniéndose así una representación significativa de lo que se va a construir de forma tal que satisfaga todos los requisitos funcionales. A continuación se muestra el diagrama de clases del diseño para la aplicación.

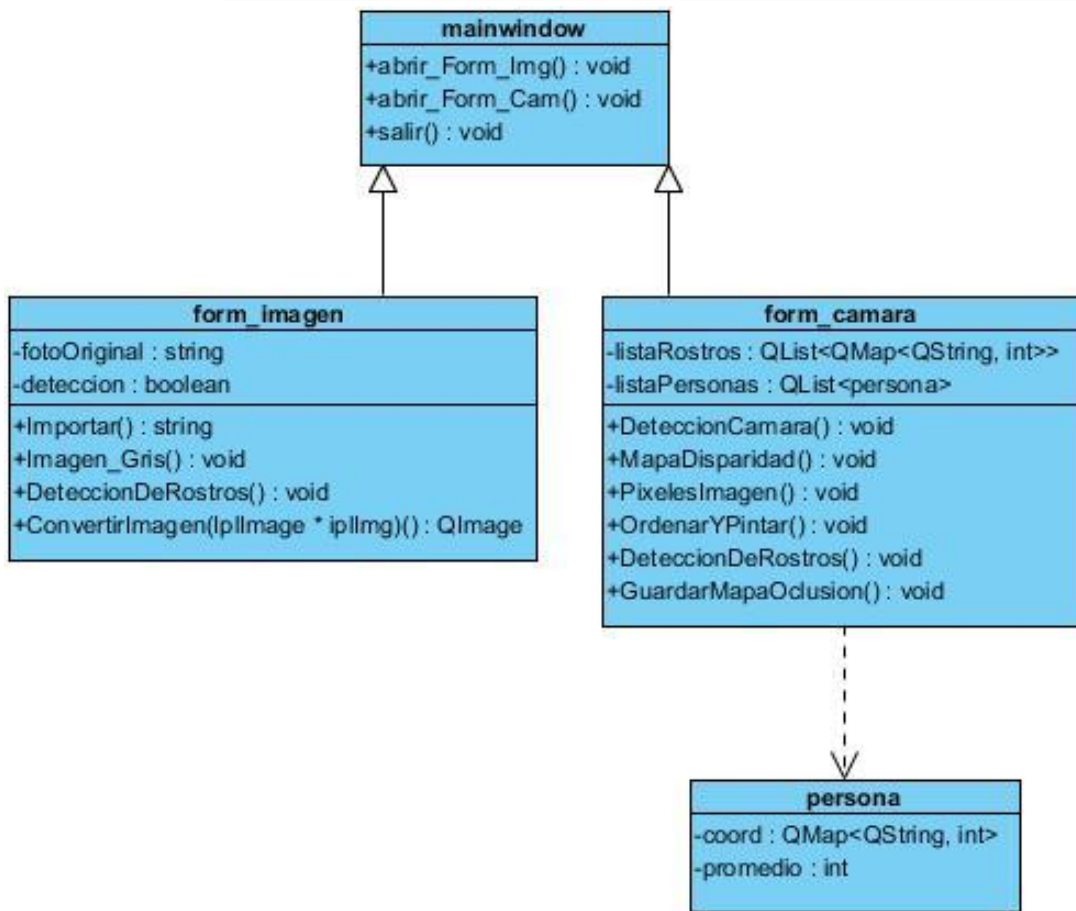


Figura 36. Diagrama de Clases del diseño.

### 3.7.2 Tarjetas CRC

Las tarjetas CRC son una técnica que admiten diseñar el sistema en conjunto, para ello se debe cumplir con tres principios: Cargo o Clase, Responsabilidad y Colaboración (CRC). Permiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos. Las tarjetas CRC permiten que el equipo completo contribuya en la tarea del diseño representando un objeto o clase de agrupamiento. La clase a la que pertenece el objeto se puede escribir en la parte superior de la tarjeta, en una columna a la izquierda se escriben las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

A continuación se muestra las tarjetas CRC generadas para el diseño de la aplicación.

Tarjeta CRC	
<b>Clase:</b> Form_Camara	
Responsabilidades	Colaboraciones
Salir a la interfaz principal.	MainWindow
Detectar los rostros en video.	Persona



<p>Obtener mapa de disparidad.</p> <p>Calcular promedio de píxeles en una región de la imagen.</p> <p>Mostrar posiciones relativas de los rostros teniendo en cuenta la profundidad.</p>	
--	--

Tabla 16. Tarjeta CRC correspondiente a la clase "Form\_Camara".

Tarjeta CRC	
<b>Clase:</b> Form_Imagen	
Responsabilidades	Colaboraciones
<p>Salir a la interfaz principal</p> <p>Importar imágenes para la detección.</p> <p>Convertir imagen a escala de grises</p> <p>Detectar los rostros en la imagen.</p> <p>Convertir la imagen a <i>Qimage</i>.</p>	<p>MainWindow</p>

Tabla 17. Tarjeta CRC correspondiente a la clase "Form\_Imagen".

Tarjeta CRC	
<b>Clase:</b> MainWindow	
Responsabilidades	Colaboraciones
<p>Salir de la interfaz principal</p> <p>Abrir la Ventana de trabajo con imágenes.</p> <p>Abrir la Ventana de trabajo con cámaras.</p>	<p>Form_Imagen</p> <p>Form_Camara</p>

Tabla 18. Tarjeta CRC correspondiente a la clase "MainWindow".

Tarjeta CRC	
<b>Clase:</b> Persona	
Responsabilidades	Colaboraciones
<p>Crear las variables necesarias para utilizar en el trabajo con la cámara.</p>	<p>Form_Camara</p>

Tabla 19. Tarjeta CRC correspondiente a la clase "Persona".

### 3.8 Patrones de Diseño

Un patrón de diseño nombra, abstrae e identifica los aspectos claves de un diseño estructurado común, que lo hace útil para la creación de diseños orientados a objetos reutilizables. En el diseño de clases propuesto se hace uso de los patrones bajo acoplamiento y alta cohesión ya que las clases son independientes y con responsabilidades bien, ejemplo de esto se puede ver en la clase "Form\_Imagen".

**Bajo acoplamiento:** Este patrón GRASP soporta el diseño de clases más independientes, no se afecta por cambios de otros componentes y las clases son de fácil comprensión por separado, así como fáciles de reutilizar.

**Alta cohesión:** Se da una alta cohesión cuando los elementos de un componente, una clase por ejemplo, colaboran para producir algún comportamiento bien definido. Una clase de alta cohesión posee un número relativamente pequeño de responsabilidades, con una importante funcionalidad relacionada y poco trabajo por hacer.

## CAPÍTULO 4. Análisis de resultados

### Introducción

En este capítulo se abordan los temas relacionados con los resultados de la implementación del prototipo basado en todo el trabajo acumulado a lo largo de los capítulos anteriores, y además se hará un análisis de las pruebas realizadas, teniendo en cuenta el cumplimiento de los objetivos propuestos.

### 4.1 Proceso de detección

A continuación se muestran los resultados en el proceso de detección de rostros haciendo uso del clasificador en cascada “*haarcascade\_frontalface\_alt*” y demás funcionalidades de biblioteca OpenCV.

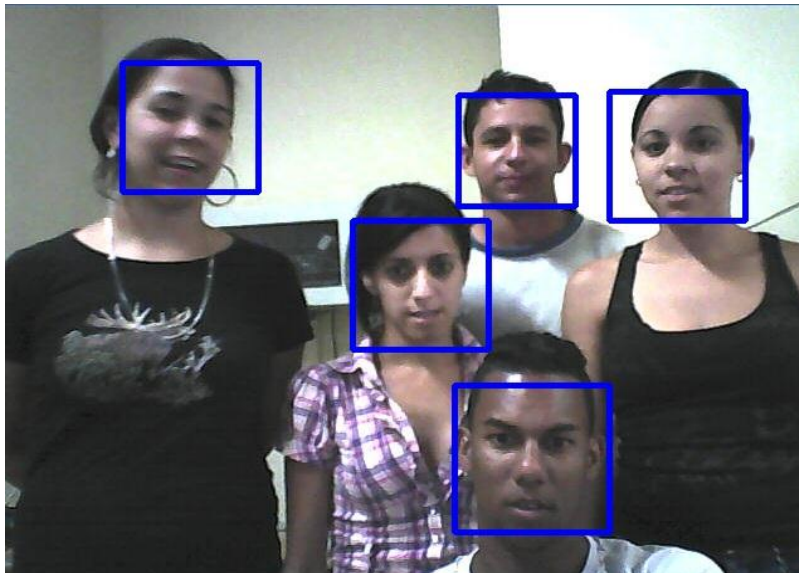


Figura 37. Detección de rostros en tiempo de ejecución.

Como se puede apreciar, cuando los rostros se encuentran de frente a la cámara (Figura 37) son detectados con eficiencia, pero cuando estos se encuentran con cierto grado de inclinación comienzan a presentar problemas, esto pasa igualmente cuando las personas se encuentran alejadas de las cámaras. El clasificador que se utiliza para la detección de rostros actúa correctamente hasta los  $20^\circ$ , una vez superado este ángulo la detección se afecta (Figura 38).



Figura 38. Imágenes con problemas de detección de rostro.

## 4.2 Obtención de mapas de disparidad

Se analizaron 2 pares de imágenes, donde cada par presentaba características diferentes y se observó que en el primer par de imágenes (Figura 39) existían buenas condiciones de iluminación, obteniéndose un mapa de disparidad acorde a lo esperado (Figura 41). Sin embargo, en el segundo par de imágenes analizadas (Figura 40) las condiciones de iluminación en la escena no eran las mejores, teniendo como resultado lo que se observa en la Figura 42.



Figura 39. Primer par de imágenes analizadas.



Figura 40. Segundo par de imágenes analizadas.



Figura 41. Mapa de disparidad GC correspondiente al primer par de imágenes.



Figura 42. Mapa de disparidad GC correspondiente al segundo par de imágenes.

### 4.3 Obtención de posiciones relativas

Los resultados de cálculo de posiciones relativas dependen de los mapas de disparidad que se obtienen, dado que el mapa de disparidad en el primer par de imágenes cumple con los requisitos para el cálculo de dichas posiciones, se obtuvo el resultado visto en la Figura 43. Con el segundo par de imágenes no se logró llegar a los mismos resultados, pues como se mencionó anteriormente, las condiciones de escena no eran las mejores.



Figura 43. Posiciones relativas de los rostros correspondientes a la Figura 39.

### 4.4 Pruebas de Aceptación

Las pruebas de aceptación son creadas en base a las HU, en cada ciclo de iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido

correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

Caso de prueba de aceptación	
<b>Código:</b> HU1_P1	<b>Historia de Usuario:</b> 1
<b>Nombre:</b> Detectar rostros en la imagen.	
<b>Descripción:</b> Evaluar que se realice la detección de rostros correspondiente.	
<b>Condiciones de ejecución:</b> Debe estar seleccionada la imagen para realizar el proceso de detección.	
<b>Entrada/Pasos de ejecución:</b> Se selecciona el botón Examinar para cargar la imagen a la cual se le quiere realizar la detección y luego se presiona el botón Detectar Rostros.	
<b>Resultados Esperados:</b> La herramienta muestra como resultado la imagen con los rostros detectados.	
<b>Evaluación de la prueba:</b> -	

Tabla 20. Caso de prueba de aceptación correspondiente a la HU 1.

Caso de prueba de aceptación	
<b>Código:</b> HU2_P1	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Detectar rostros en video.	
<b>Descripción:</b> Evaluar que el sistema cargue los dispositivos de captura y que además se realice la detección de rostros en video correspondiente en tiempo de ejecución.	
<b>Condiciones de ejecución:</b> Se debe comprobar que los dispositivos de captura estén conectados e instalados correctamente en la computadora.	
<b>Entrada/Pasos de ejecución:</b> Se selecciona el botón Detectar Rostros correspondiente y se ejecuta dicha funcionalidad.	
<b>Resultados Esperados:</b> La herramienta muestra como resultado la detección de rostros en tiempo de ejecución.	



**Evaluación de la prueba: -**

Tabla 21. Caso de prueba de aceptación correspondiente a la HU 2.

Caso de prueba de aceptación	
<b>Código:</b> HU3_P1	<b>Historia de Usuario:</b> 3
<b>Nombre:</b> Obtener mapa de disparidad.	
<b>Descripción:</b> Evaluar que se realice la obtención del mapa de disparidad correspondiente.	
<b>Condiciones de ejecución:</b> Se debe asegurar que se obtengan el par de imágenes para realizar los cálculos correspondientes.	
<b>Entrada/Pasos de ejecución:</b> Se selecciona el botón Mapa Disparidad en el cual se ejecuta la funcionalidad.	
<b>Resultados Esperados:</b> La herramienta muestra el mapa de disparidad obtenido y las imágenes con las cuales se obtuvo.	
<b>Evaluación de la prueba: -</b>	

Tabla 22. Caso de prueba de aceptación correspondiente a la HU 3.

Caso de prueba de aceptación	
<b>Código:</b> HU4_P1	<b>Historia de Usuario:</b> 4
<b>Nombre:</b> Calcular posiciones relativas de los rostros detectados.	
<b>Descripción:</b> Evaluar que se realice el cálculo de posiciones relativas de los rostros en la imagen.	
<b>Condiciones de ejecución:</b> Se debe asegurar que se haya realizado el proceso de mapa de disparidad, pues a través de este es que se realizan los cálculos correspondientes.	
<b>Entrada/Pasos de ejecución:</b> Se selecciona el botón Posiciones Relativas el cual ejecuta la funcionalidad.	
<b>Resultados Esperados:</b> La herramienta muestra como resultado la imagen con sus posiciones relativas en un cuadrado alrededor del rostro y guarda un fichero de texto con la información de profundidad.	
<b>Evaluación de la prueba: -</b>	

Tabla 23. Caso de prueba de aceptación correspondiente a la HU 4.



## Conclusiones

Con la realización de este trabajo y en el transcurso de la investigación se arriban a las siguientes conclusiones:

- El algoritmo de Viola y Jones permite tomar sus resultados como base para detectar los rostros de las personas en una escena.
- Siguiendo la misma técnica de los mapas de oclusión, se puede representar la posición relativa de personas en una escena.
- Con el uso de cámaras web se desarrolló una aplicación que permite realizar la detección de rostros, ya sea con imágenes o a partir de la captura de video en tiempo de ejecución y representar además sus posiciones relativas en la escena.

## Recomendaciones

---

A continuación se enuncian algunas recomendaciones orientadas a trabajos futuros para lograr resultados satisfactorios:

- Utilizar cámaras con características lo más similares posibles (preferentemente iguales) para lograr una buena calibración de las mismas.
- Entrenar los clasificadores en cascada o implementar alguna de las mejoras propuestas en la bibliografía para mejorar el proceso de detección.
- Incluir el análisis de la iluminación o el color en la escena para lograr nuevos resultados.

---

## Bibliografía

1. **Dragolici, Andrei Bogdan Dragolici.** *Detección, localización e identificación biométrica de caras en imágenes: Métodos y evaluación en el marco NIST-MBGG.* 2010.
2. **G. Yang, T.S. Huang.** *Human Face Detection in a Complex Background.* 1994.
3. **M.H. Yang, N. Ahuja.** *Detecting Faces in Images: A Survey.* 2002.
4. **J. Ohya, R. Nakatsu, J. Tang, S. Kawato.** *Locating human faces in a complex background including non-face skin colors.* 2000.
5. **R.Lienhart, J.Maydt.** *An Extended Set of Haar-Like Features for Rapid Object Detection.* 2002.
6. **Y.Freund, R.Schapire.** *Experiments with a new boosting algorithm International Conference on Machine Learning.* 1996.
7. **C.Garcia, G.Tziritas.** *Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis.* 1999.
8. **A.L.Yuille, P.W.Hallinan, D.S.Cohen.** *Feature extraction from faces using deformable templates.* 1992.
9. **Sinha, P.** *Object Recognition via Image Invariants: A case Study, Investigative Ophthalmology and Visual Science.* 1994.
10. **T. V. Pham, M. Worring, A. W. M. Smeulders.** *Face detection by aggregated Bayesian network classifiers.* 2001.
11. **D. Roth, M.-H. Yang, N. Ahuja.** *A SNoW-Based Face Detector.* 2000.
12. **K. N. Ogle, W.B.** *Binocular Vision.* Philadelphia : s.n., 1950.
13. **Colunga, Mario Gonzalo Chirinos.** *Visión Artificial: Percepción de Profundidad.* 2004.
14. **Helmholtz, H. von.** *Physiological optics.*
15. **Julesz, B.** *Foundations of Cyclopean Perception.* London : s.n., 1971.
16. **David A Forsyth, Jean Ponce.** *Computer Vision: A Modern Approach.*
17. **Faugeras, O.D.** *Three-dimensional computer vision: a geometric viewpoint.* 1993.
18. **Martin, Santiago.** *Ingeniería Inversa en el diseño para la fabricación aditiva mediante técnicas fotográficas.* 2009.
19. **Mursuli, Adrian Fonseca.** *Adquisición de modelos geométricos a partir de objetos del mundo real.* 2011.
20. **Rodríguez, L. S.** *Obtención de mapas de profundidad densos mediante visión activa por movimiento controlado de una cámara.* 2003.
21. **Garcia, J.** *Integración de objetos sintéticos en imágenes y video real.* 2007.
22. **Jimenez, J. G.** *Visión por Computador.* 1999.
23. **Zhang, A.** *A Flexible New Technique for Camera Calibration.* 1999.

24. **Tsai, R. Tsai.** *"A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses.* 1987.
25. **Bary Gradsy, A. K.** *Learning OpenCV.* s.l. : O'Reilly Media, 2008.
26. **José H. Canós, Patricio Letelier y M<sup>a</sup> Carmen Penadés.** *Metodología Agiles de Desarrollo de Software.*
27. **P. Viola, M. J. Jones.** *Robust Real-Time Face Detection.* 2004.
28. **Graffigna, Juan Pablo, Romero, Luis Eduardo y R, Raúl.** *Evaluación de métodos para la obtención del mapa de disparidad en sistemas de visión estéreo.*
29. **Ivar Jacobson, Grady Booch y Rumbaugh, James.** *Proceso Unificado de Desarrollo de Software.*
30. **Letelier, Patricio.** *Métodologías ágiles para el desarrollo de software:eXtreme Programming (XP).*
31. **Pressman, Roger S.** *Ingeniería del Software: Un enfoque práctico.*
32. **García, Javier Y.** *Tracking de personas a partir de Vision Artificial.*

## Glosario de términos

### C

**C++:** Lenguaje de programación, la intención de su creación fue extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos.

### G

**GRASP:** Patrones generales de software para asignación de responsabilidades (del inglés, *General Responsibility Assignment Software Patterns*).

### H

**HSV:** Define un modelo de color en términos de sus componentes constituyentes (del inglés, *Hue, Saturation, Lightness*, en español, Matiz, Saturación, Luminosidad).

### N

**NCC:** Método para calcular un mapa de disparidad, representa la correlación cruzada normalizada (del inglés, *Normalized Cross Correlation*).

### R

**RGB:** Es la composición del color en términos de la intensidad de los colores primarios de la luz (del inglés *Red, Green, Blue*, en español rojo verde azul).

**ROI:** región de interés (del inglés, *Region Of Interest*).

### S

**SNOW:** Es una arquitectura de aprendizaje que se adapta para el aprendizaje en la presencia de un gran número de fuentes de información (del inglés, *Sparse Network of Windows*, en español Red Dispersa de Aventa).

**SSD:** Método para calcular un mapa de disparidad, representa la suma de diferencias al cuadrado (del inglés, *Sum of Squared Differences*).

**SAD:** Método para calcular un mapa de disparidad, representa la suma de diferencias absolutas (del inglés, *Sum of Absolute Differences*).

### U

**UML:** Lenguaje Unificado de Modelado (del inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.