

PROYECTO FIN DE CARRERA

DETECCIÓN Y SEGUIMIENTO DE ROSTROS



UNIVERSIDAD CARLOS III DE MADRID

EDWIN ARTURO VEGA AQUINO

2011

Para mis padres y hermanas, por todo el apoyo que me brindaron y lo siguen haciendo hasta el día de hoy.

Índice general

Agradecimientos.....	4
1. Capítulo I: Introducción.....	5
1.1 Prólogo.....	7
1.2 Motivación.....	8
1.3 Objetivos.....	11
2. Capítulo II: Técnicas de Detección de Rostros.....	12
2.1 Historia.....	14
2.2 Problemas habituales en la detección de rostros.....	15
2.2.1 Detección de un rostro humano.....	15
2.2.2 Localización de componentes faciales.....	16
2.2.3 Estimación de pose.....	17
2.2.4 Seguimiento de rostros en un video.....	17
2.2.5 Análisis de la expresión facial.....	17
2.2.6 Extracción de información.....	18
2.3 El estado del arte en detección de rostros.....	18
2.3.1 Métodos descendentes basados en conocimiento.....	18
2.3.2 Métodos ascendentes basados en invariantes.....	19
2.3.3 Métodos basados en patrones predefinidos.....	21
2.3.4 Métodos basados en apariencia.....	22
2.4 Eigenface.....	23
2.4.1 Cálculo del "espacio de rostros".....	24
2.4.2 Proyección de nuevas imágenes en el "espacio de rostros".....	24
2.4.3 Cálculos matemáticos.....	24
2.4.4 Problemas del Eigenface.....	26
2.5 Clasificadores en Cascada.....	26
2.5.1 Cálculo de la imagen integral.....	26
2.5.2 Extracción de características: Filtros Base Haar.....	26
2.5.3 Clasificador: AdaBoost.....	27
2.6 Seguimiento de rostros en video.....	28
2.6.1 El problema del seguimiento.....	28
2.6.2 Clasificación según el modelo de seguimiento.....	31
2.6.3 Clasificación según la estrategia de búsqueda.....	32
2.6.4 Mecanismos de predicción.....	32
2.6.5 Seguimiento de caras basado en color y otras características.....	33
2.6.6 Seguimiento de caras basado en apariencia.....	34

3. Capítulo III: Visión Artificial.....	36
3.1 Visión Artificial.....	38
3.1.1 Por qué el uso de la visión artificial.	40
3.1.2 Componentes de un sistema de visión artificial.....	46
3.1.3 En la actualidad.	49
3.1.4 Justificación de la inversión en visión artificial.....	56
3.2 Concepto de las OpenCV.	58
3.2.1 Representación Gráfica.	60
3.2.2 La competencia.	61
4. Capítulo IV: Seguimiento.....	62
4.1 Seguimiento.	64
4.1.1 Primera solución.	65
4.1.2 Segunda solución.	68
4.1.3 Mejoras con esta propuesta.	74
4.2 Descripción general del proyecto.	76
5. Capítulo V: Resultados Experimentales.....	78
5.1 Primer Experimento: video de 320 x 240.....	80
5.2 Segundo Experimento: video de 640 x 480.....	82
5.3 Tercer Experimento: cambios de iluminación.....	85
5.4 Cuarto Experimento: velocidad.	87
6. Capítulo VI: Conclusiones.....	90
6.1 Conclusiones.	92
6.1.1 Ventajas.	92
6.1.2 Limitaciones.	93
6.2 Trabajos Futuros.	94
7. Capítulo VII: Presupuesto.....	95
8. Capítulo VIII: Bibliografía.....	98

Agradecimientos

Agradecer a mi familia, que me inculcaron la educación y el bien estar por encima de todo, el seguir mis principios y la superación personal y profesional. También agradecer a José María Armingol y Arturo de la Escalera por implicarse en éste proyecto.

CAPITULO I

Introducción



La ciencia puede descubrir lo que es cierto, pero no lo que es bueno, justo y humano.
Marcus Jacobson

1.1 Prólogo

Éste proyecto ha sido pensado para poner nuestro grano de arena a los grandes avances tecnológicos de éstos últimos tiempos; así pues, contribuir de ésta manera a proyectos futuros con la esperanza de que muy pronto; ya sea basada en ésta técnica, en otras o tal vez en un híbrido de ellas, puedan dar solución a los diversos problemas que hoy se plantean, para el seguimiento de rostros.

En el primer capítulo, encontraremos los principales motivos por el cual dedicamos nuestro esfuerzo a la investigación y al desarrollo de nuevas técnicas para la implementación en la detección y seguimiento de rostros y objetos. Así como también se describe el objetivo de éste proyecto.

En el segundo capítulo, explicamos las principales técnicas para la detección y seguimiento de rostros que se han ido desarrollando en el transcurso del tiempo, las principales ventajas y desventajas de cada una de ellas, así como también sus diferencias. También hacemos hincapié a una de las más populares técnicas que es la Eigenface. Profundizaremos en explicar los conceptos acerca de los clasificadores en cascada y los filtros de base haar, ya que son la base de nuestro proyecto.

En el tercer capítulo, haremos mención a las librerías de la OpenCV ya que son fundamentales para la realización de nuestro proyecto, así como también profundizaremos en los conceptos y la actualidad de la visión por ordenador.

En el cuarto capítulo, explicaremos con lujo de detalles el proyecto que se ha realizado, haciendo hincapié en las técnicas utilizadas para el seguimiento del rostro.

En el quinto capítulo, se podrá apreciar los resultados experimentales que hemos obtenido, al momento de probar nuestro programa con diferentes clases de videos.

En el sexto capítulo, tomamos en cuenta los resultados experimentales y sacamos nuestras conclusiones finales del proyecto. También comentamos sobre los posibles trabajos futuros que se podría realizar con éste proyecto.

En el séptimo capítulo, damos un presupuesto estimado acerca del costo económico del proyecto.

En el octavo y último capítulo, como en todo proyecto, terminamos con la bibliografía, no es más que un listado de libros, artículos y manuales que han sido consultados y que gracias a su ayuda han facilitado la realización de este proyecto.

1.2 Motivación

En los últimos años la tecnología ha crecido de un modo espectacular, éste incremento ha hecho posible que influya en nuestras vidas, casi sin darnos cuenta de ello; es verdad que, como todo en esta vida, la tecnología trae ventajas y desventajas.

No es mi propósito hablar de las desventajas ya que a mi entender pueden ser hasta cierto punto controladas y reducir el riesgo que conllevan. Lo que quiero destacar es la gran importancia que tiene y la diversidad de formas en las que nos pueden ayudar.

Si orientamos lo dicho anteriormente a este proyecto, veremos que su aplicación también es muy variada y aquí citamos algunos ejemplos:

a) Entretenimiento:

Para aquellos que les gustan los deportes, el famoso ojo de halcón por ejemplo, presenta una técnica de seguimiento de la pelota, el cual nos permite saber si un punto es válido o no. También podemos aplicar estos conocimientos para datos estadísticos, por ejemplo en el fútbol, cuando queremos saber la posesión del balón de un determinado equipo, el fuera de juego y diversos datos estadísticos más. También es posible mejorar la detección de rostros en cámaras fotográficas y video cámaras.

En la figura 1 vemos el funcionamiento del ojo de halcón en el momento en que se ha producido un servicio de uno de los jugadores.

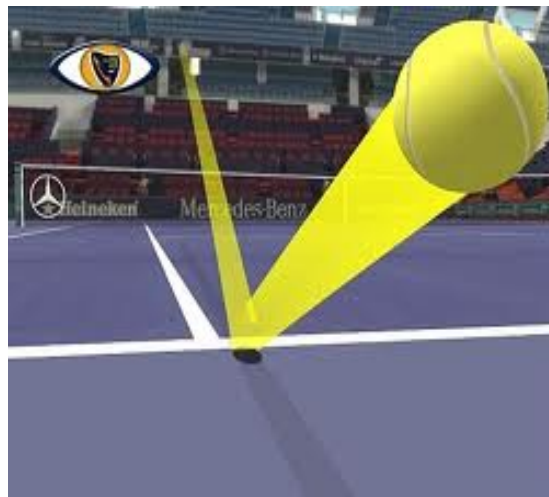


Figura 1: Ojo de halcón

b) Seguridad:

Después del 11 de setiembre, es una de las ramas más importante en donde la tecnología ha ganado terreno. El seguimiento de rostros de potenciales terroristas, o la detección de un objeto sospechosamente abandonado son las principales aplicaciones, un ejemplo de ésta aplicación la vemos en la figura 2 en el cual se puede apreciar una maleta sospechosamente abandonada.



Figura 2: objeto sospechoso

c) Hardware:

Mediante la detección de un rostro, éste puede mandar una señal a la cámara y hacer que la cámara se mueva, y de esta manera no perder de vista a la persona deseada, esto es la interacción del software con hardware, para una mejor optimización. En la figura 3 se muestra el funcionamiento de las cámaras de vigilancia.



Figura 3: Cámaras de Seguridad

d) Seguridad Vial:

Los accidentes de tráfico son muy frecuentes en nuestra sociedad, y aunque las leyes sobre el alcohol y demás se han vuelto más severas, no han reducido la tasa de accidentes como se quisiera.

El trabajo hacia un coche “inteligente” que sea capaz de interpretar las señales de tráfico y actuar según su interpretación-programación es uno de los campos en los que se está trabajando actualmente, si es que no existen ya los primeros prototipos en el laboratorio. En la figura 4 podemos apreciar las distintas aplicaciones de una cámara instalada sobre un vehículo.



Figura 4: Cámaras de Seguridad

e) Visión artificial:

La visión artificial ha interactuado con la biomedicina, para intentar devolver la visión a pacientes con ceguera, eso ha sido posible gracias a los avances tecnológicos que ha experimentado esta rama de la ciencia. En la figura 5 vemos un dispositivo que intenta devolver la visión a los que sufren de ceguera.



Figura 5: Visión Artificial

1.3 Objetivo

El objetivo fundamental del proyecto es el seguimiento del rostro, una vez detectado el rostro en la imagen mediante la función `cvHaarDetectObjects` (de las librerías de las OPENCV) procederemos a desactivar ésta función para luego continuar con el seguimiento mediante la estimación de movimiento.

Cada cierto tiempo se hará un refresco para evitar que los errores se acumulen en el tiempo. Se podría decir que es una optimización de la función `cvHaarDetectObjects`, o un complemento de ésta; ya que, al trabajar conjuntamente, la vuelve más rápida, alcanzando en ciertos casos específicos, la iteración a tiempo real.

La idea fundamental de éste proyecto; es decir, el funcionamiento del método reconoce de la clase Seguimiento, está inspirada en la codificación de video MPEG 1, el cual utiliza imágenes intra (I) e inter (P).

Las imágenes intra son aquellas que contienen toda la información y en la cual se produce la reducción de la resolución espacial para la codificación de video; además se refrescan cada cierto tiempo. Su análogo en nuestro proyecto es la información que nos facilita la función `cvHaarDetectObjects`.

Las imágenes inter o predictivas, tienen sólo la información necesaria y son usadas para reducir la resolución temporal del video mediante la estimación de movimiento, su análogo en nuestro proyecto es el método reconoce de la clase Seguimiento.

Para la realización del proyecto se ha utilizado Visual Studio 2008 como entorno de desarrollo, C++ como lenguaje de programación y las librerías de las OpenCV, para obtener el máximo rendimiento en el seguimiento de rostros, como muestra la siguiente imagen.



Figura 5: Seguimiento

CAPITULO II

Técnicas de detección de rostros



Los errores ajenos proporcionan los medios para descubrir la verdad.
John Selden

Si se puede imaginar... se puede programar.

2.1 Historia

A lo largo de éste capítulo se verán conceptos que han sido adquiridos gracias a gran parte de lo expuesto en el octavo capítulo que pertenece a la bibliografía, por tanto si se desea entrar con más profundidad, se pueden consultar los textos: [3], [4], [5], [6], [7], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [22], [23], [24] y [25].

- En 1888; aunque pueda parecer sorprendente, Francis Galton fue el primero en proponer un método sobre clasificación de caras. Éste recogía perfiles faciales como curvas, encontrando su norma, después se clasificaban perfiles por sus desviaciones con respecto a la norma. El resultado era un vector que podía ser comparado con otros vectores de la base de datos.
- En 1960; se introdujeron sistemas semiautomáticos que hacían marcas en las fotografías para localizar las marcas principales: ojos, orejas, nariz y boca. Las distancias y radios se calculaban a través de las marcas para construir un sistema de referencia y poder comparar los datos.
- En 1970; Goldstein, Harmon y Lesk crearon un sistema con 21 marcadores que incluían color del pelo y grosor de labios. Sus pruebas eran también difíciles de automatizar porque muchas de estas medidas se tomaban a mano.
- Pocos años después Fisher y Elschlagerb introdujeron un sistema más autónomo que utiliza plantillas para medir los rasgos de diferentes partes de la cara, con las cuales construían un mapa global. Tras una continuada investigación resultó que las medidas no contenían suficientes datos únicos como para representar una cara de adulto.
- Otra aproximación intenta clasificar la cara humana usando una combinación de la gama de gestos como un juego de marcadores a identificar. En general se pone en práctica usando reconocimiento de modelo 2D y principios de redes neuronales. La mayoría de veces esta técnica requiere un enorme número de caras a entrenar para alcanzar una exactitud respetable.
- El primer método completamente automatizado, compara las caras en un modelo genérico de rasgos esperados y creaba una serie de patrones para una imagen. Ésta estrategia era principalmente estadística, se basaba en histogramas y en el valor de la escala de grises.

- En 1988; los trabajos, en un primer momento, de Kirby y Sirovich; y luego de Turk y Pentland basados en el método de Eigenface abrieron un camino al que ha ido contribuyendo mucha gente.
- En 1990; incremento el interés por el reconocimiento de caras automático se han sumado un creciente número de grupos de investigación. Apoyándose también en las mejoras técnicas que han supuesto el desarrollo de las redes neuronales, análisis wavelet, infografía y visión por ordenador.
- Actualmente gracias al 11S, ha incrementado el interés por el desarrollo y mejora de esta tecnología, aquí en España el 11M produjo los mismos efectos ya que demostró que el terrorismo es un problema mundial. Desde entonces las más importantes universidades de España han dedicado gran parte de su tiempo en promover la investigación de todas estas tecnologías con el fin de brindarnos un mundo más seguro.



Figura 6: Robot

2.2 Problemas habituales en la detección de rostros

2.2.1) Detección de un rostro humano

Aunque pueda parecer increíble, en algunas ocasiones, incluso para las propias personas, nos es fácil reconocer un rostro humano, pongamos por ejemplo aquella imagen de Marte (figura 7) donde se podía apreciar un rostro un tanto humanoide por llamarlo de alguna manera y que dio la vuelta al mundo.



Figura 7: Marte

Un segundo ejemplo de lo que expuesto anteriormente es el célebre sudario de Turín (figura 8) en el cual se cree que aparece el rostro de Jesús. ¿Se puede catalogar a estas imágenes como rostros humanos? o son simple coincidencias de luces y sombras o de otro efecto natural que lo provoca. Obviamente la finalidad de éste proyecto no es abrir un debate religioso ni mucho menos, sólo se quiere plasmar las dudas surgieron y que seguirán surgiendo en dichos aspectos.



Figura 8: Sudario de Turín

2.2.2) Localización de componentes faciales

Una vez que se ha detectado el rostro, el siguiente objetivo es la localización de sus componentes, como por ejemplo, su posición, la boca, los ojos, y las demás que se crean convenientes. A diferencia de la detección de rostros, en donde lo importante es una buena discriminación rostro/no-rostro, en éste caso lo esencial es la precisión de los resultados, que puede verse influida por la mala resolución de entrada. Como ejemplo de lo expuesto se puede apreciar en la figura 9.



Figura 9: Componentes faciales

2.2.3) Estimación de pose

Se trata de saber la posición de la cabeza del individuo en relación a la cámara, obviamente es un problema de naturaleza tridimensional, para ello se podría usar modelos 3D, aunque también se puede trabajar con 2D. Es un problema que cada técnica resuelve de distintas maneras, y que influye mucho en el buen funcionamiento del programa ya que se podría catalogar como un paso crítico para la estimación, debido a que los errores que provoque serán arrastrados en el tiempo en futuras comprobaciones y con ellos reducirá el éxito del objetivo.

2.2.4) Seguimiento de rostros en un video

Su principal propósito es encontrar la variación de posición, forma y orientación del rostro a lo largo de una secuencia de video. La capacidad humana para el reconocimiento de rostros es asombrosa, incluso en condiciones desfavorables, sin embargo ésta capacidad puede ser muy subjetiva; ya que, puede ser fácilmente confundida con simples alteraciones en la imagen, lo que comúnmente se llaman juegos de imágenes.

2.2.5) Análisis de la expresión facial

El problema consiste en clasificar la expresión del sujeto en un número discreto y predefinido de clases; aunque, otros prefieren buscar una estimación graduada del estado de activación de los músculos faciales.

En ambos casos, sería conveniente diferenciar entre el análisis del gesto (boca abierta o cerrada, cejas subidas, ojos abiertos, etcétera) y la interpretación emocional (triste, contento, enfadado, etcétera), éste último mucho más complejo ya que se requiere una interpretación del contexto y consideraciones sociológicas, inherentes al ser humano y que por ejemplo un robot sería incapaz de detectar.



Figura 10: Gestos

2.2.6) Extracción de información

El resultado podría estar orientado a mejorar las imágenes para una etapa posterior de reconocimiento biométrico. Aquí lo que queremos saber es si un individuo tiene gafas, barba, su sexo, su raza, la edad, etc.

2.3 El estado del arte en detección de rostros

En la actualidad, el campo de la detección facial parece haber alcanzado cierta madurez con la existencia de varias implementaciones bastante fiables; no obstante, algunas cuestiones siguen aún pendientes. Podemos distinguir cuatro grandes categorías de métodos de detección de rostros, basados en conocimientos, en propiedades invariantes, en patrones predefinidos y en apariencia.

2.3.1) Métodos descendentes basados en conocimiento

El problema con ésta técnica es como crear un conjunto óptimo de reglas. Si son muy detallados se restringe la capacidad de generalización y si las reglas son muy generales incrementa el número de falsas detecciones. Esto se debe a que el buen funcionamiento dependerá también del conocimiento del investigador y de su experiencia para delimitar las reglas.

La estrategia de búsqueda descendente (top-down), propone un sistema de reglas a tres niveles, con resoluciones crecientes en las imágenes. Un ejemplo claro es el trabajo de Yang y Huang [15].

1. Primer nivel; usa la resolución más baja, se busca una distribución de intensidades, dada por un patrón de 5x5 píxeles, creado de forma natural.
2. Nivel intermedio; se define un criterio basado en detección de bordes sobre los candidatos resultantes del paso anterior.
3. Aquellos candidatos que pasen ambas criterios, se someten a otro test de mayor resolución con reglas orientadas a la distinción de cada uno de los componentes faciales por separado.

Los resultados experimentales de éste método, logra una capacidad de detección del 83% sobre un conjunto de 60 imágenes, pero tiene un 47% de falsas alarmas por imagen.

Los métodos dentro de ésta categoría se ven sujetos a las enormes dificultades de expresar en un conjunto reducido de variables y/o reglas, la infinidad de cambios de apariencia que experimenta el rostro humano; por ello, se puede decir que es una línea abandonada.

2.3.2) Métodos ascendentes basados en invariantes

Se da por hecho que deben existir características de bajo nivel, fáciles de calcular pero invariantes frente a un gran número de situaciones complejas. Frente a los algoritmos basados en conocimiento, el procesamiento suele seguir un camino ascendente: primero se extraen características de bajo nivel individualmente, y luego se agrupan utilizando descriptores estadísticos.

a) Bordes:

La extracción de bordes del rostro ha sido una de las propiedades invariantes más utilizadas, tanto externos como internos, luego eran sometidos a un análisis de forma y posición. Para la extracción de bordes se utilizan distintos métodos, como el operador de Canny, convoluciones con segunda derivada de la gaussiana, gradiente de Sobel, etc.

En esta ocasión se ha realizado la gradiente de Sobel a imagen Lena (figura 11) con el propósito de apreciar el buen funcionamiento de ésta técnica, el cual nos permite obtener los bordes de una imagen. Dependiendo del grado y de la técnica utilizemos, la detección de bordes será mejor o peor.



Figura 11: Gradiente

b) Descriptores de textura:

Generalmente se definen tres tipos de texturas: piel, pelo y otras. La clasificación de los bloques se realiza mediante redes neuronales, y para la detección de caras un esquema que mida la ocurrencia de pelo y piel. También se puede añadir información de color para aumentar la relevancia del color anaranjado.

c) Color:

Si bien es cierto que las imágenes en blanco y negro permiten una representación básica de la imagen para extraer propiedades, el color es una herramienta mucho más poderosa, entre otras cosas porque nos brinda mayor información, como por ejemplo el color de la piel humana (en sus diferentes variantes), esto ha permitido desarrollar algunas técnicas que detectan la raza (algunas representadas en la figura 12), para este tipo de técnicas se suele trabajar con un RGB normalizado.



Figura 12: Estilos

Los métodos de detección basados en color presentan varios puntos débiles, en primer lugar la variación de los matices entre diferentes sistemas de adquisición y fuentes de iluminación, además de la perturbación por ruido y por compresión, es tan grande que hace inviable fijar un espacio universal para todos los colores de piel humana y la clasificación piel/no piel, no sería efectiva; incluso si éste método fuera perfecto, no es suficiente para detectar rostros, puesto que también encontraría manos, brazos, pies, etc.

d) Combinación de color, forma y proyecciones:

Dentro de los métodos de rostros basados en invariantes, los más exitosos se encuentran entre los que hacen uso de diferentes características.

1. Clasifica los píxeles en piel/no piel usando un modelo de color.
2. Buscar rostros candidatos mediante agrupación de componentes conexos o con clustering de los píxeles de color de piel.
3. Aplicar una etapa de verificación de candidatos, basada en comprobar la existencia de ojos, nariz y boca.

Existen otros muchos métodos ascendentes que combinan otros tipos de características que se aplican junto al color, como la estructura del rostro, los momentos geométricos, la simetría, y la profundidad.

2.3.3) Métodos basados en patrones predefinidos

Se apoyan en modelos predefinidos manualmente, se distinguen dos categorías: los que manejan patrones predefinidos, y los que permiten patrones deformables.

1) Patrones predefinidos:

El uso de operadores de bordes es una de las elecciones más frecuentes; es decir, no se crean modelos de las imágenes de intensidad, sino de las imágenes de bordes. Se realiza un proceso de agrupación de contornos y se buscan tres curvas que puedan corresponder a los bordes de la cara, el porcentaje de detección se estima en un 70%.

Otra forma de aplicar patrones basados en bordes es dividir el rostro en bloques, donde cada uno tiene información de luminosidad y nivel de bordes del mismo.

2) Patrones deformables:

Permiten cierta flexibilidad en el ajuste de un modelo de rostro. Estos métodos definen una función de energía, que mide el grado de ajuste en cada momento. Básicamente, la energía consta de dos componentes: uno interno y otro externo. El externo tiende a ajustar al máximo el modelo a la instancia actual, mientras que el interno hace que el ajuste actual mantenga una forma coherente y compacta.

2.3.4) Métodos basados en apariencia

Las técnicas expuestas anteriormente, se centran en la detección de una única cara en las imágenes, con una buena resolución y en la mayoría de los casos, mirando de frente.

Las técnicas basadas en apariencia intentan resolver estas limitaciones y es una de las más exitosas. Básicamente, la idea subyacente consiste en reducir la detección a un problema de clasificación cara/no cara, que se repite de forma exhaustiva para todas las posibles subregiones y con todas las escalas admitidas.

La cuestión fundamental es entrenar un clasificador binario a partir de un conjunto amplio de ejemplos de caras y de no caras. Por ello a veces estas técnicas se denominan también basadas en aprendizaje.

Los resultados prácticos de estos métodos son:

- Son capaces de manejar situaciones complejas, con imágenes de escasa resolución, baja calidad y un número arbitrario de rostros por imagen.
- Pueden presentar un riesgo de sobre-ajuste a los datos de aprendizaje, de manera que el entrenamiento del clasificador se convierte en el aspecto clave.
- El funcionamiento exhaustivo del proceso implica un elevado coste computacional que en general será uno de los principales inconvenientes.

1. Búsqueda multiescala:

El primer elemento del proceso es una búsqueda exhaustiva multiescala. A partir de la imagen de entrada, se obtiene una pirámide de imágenes con diferentes resoluciones. Se define un factor de escala, de manera que cada imagen en la pirámide será cada vez menor según el factor de escala.

2. Pre-procesamiento y foco de atención:

Antes de la clasificación, muchos métodos incluyen una etapa de pre-procesamiento de las ventanas obtenidas. Este paso puede tener dos propósitos:

- Descartar regiones en las que claramente no existe una cara. Para ello se pueden aplicar heurísticas sencillas basadas en bordes para eliminar regiones que sean muy uniformes.

- Mejorar las imágenes de las ventanas de cara a la etapa posterior de clasificación. Fundamentalmente, se pretende dos cosas: aumentar el contraste de las sub-imágenes (mediante normalización del brillo), y reducir el posible efecto de sombras no uniformes.

3. Clasificación rostro/no rostro:

Estos clasificadores, deben ser capaces de modelar la compleja forma del rostro y al mismo tiempo tener una ejecución extremadamente rápida. El problema se formula dentro de un marco probabilístico, donde las ventanas que encierran al rostro, se consideran como variables aleatorias "x" y las clases rostro/no rostro como dos distribuciones de probabilidad; la dificultad está en la elevada dimensionalidad de "x" y en los múltiples modos de variación posibles de la clase rostro.

Para resolverlo, se han aplicado distintos métodos de clasificación: redes neuronales, discriminantes lineales de Fisher, máquinas de vectores de soporte, reducción a sub-espacios lineales, combinación de clasificadores elementales (como el método AdaBoost), etc.

4. Agrupación de candidatos y post-procesamiento:

Como resultado de la clasificación de las ventanas, tenemos un conjunto de posibles regiones del rostro. Muchas de ellas pueden estar asociadas a un mismo rostro, y alguna de ellas pueden ser falsas detecciones. El último paso de los detectores basados en apariencia consiste en agrupar las regiones con un alto solapamiento y eliminar a los candidatos menos fiables.

2.4 EIGENFACE

El principio básico de las Eigenface es la extracción de características, su objetivo es capturar las variaciones existentes entre un conjunto de imágenes de rostro.

En términos matemáticos, se desea encontrar las componentes principales de la distribución de rostros, o los autovectores de la matriz de covarianza de un conjunto de imágenes de rostros, considerando que una imagen es un punto o vector en un espacio dimensional muy alto. Los autovectores son ordenados, cada uno representando una diferente cantidad de variación entre los rostros.

La técnica para el reconocimiento de rostros consiste en dos etapas:

2.4.1) Cálculo del "espacio de rostros"

- Analiza las componentes principales en el conjunto de imágenes de entrenamiento, proyecta cada imagen en el espacio calculado y crea clases de clasificación.
- Conseguir un conjunto de imágenes de rostros (conjunto de entrenamiento).
- Calcular los componentes principales del conjunto de entrenamiento, guardando solo "M" de ellos correspondientes a los autovectores más altos. Estos "M" componentes principales definen el "espacio de rostros". Al ir aumentando más rostros, los componentes pueden ser actualizados o recalculados.
- Calcular la distribución correspondiente a cada individuo, proyectando la imagen de su rostro en el "espacio de rostros".

2.4.2) Proyección de nuevas imágenes en el "espacio de rostros"

- Para la comparación con cada clase en el sistema y su posterior clasificación.
- Calcular un conjunto de pesos de la imagen de entrada al proyectarla en cada uno de los componentes principales.
- Determinar si la imagen es un rostro verificando su cercanía con el "espacio de rostros".
- Si es un rostro, clasificar el patrón de pesos como conocido o desconocido.
- Actualizar los componentes principales y patrones de pesos.

2.4.3) Cálculos matemáticos

El objetivo de éste apartado, no es la de calcular cada uno de los aspectos matemáticos que intervienen en la técnica de Eigenface, sino la de explicar los pasos y cálculos más importantes.

a) Inicialización:

- Tenemos un conjunto de rostros I_1, I_2, \dots, I_M . Cada I_i es una matriz $N \times N$.
- Representamos cada matriz I_i como vector Γ_i . Cada Γ_i es un vector $N^2 \times 1$.
- Calculamos el promedio del conjunto de vectores Γ_i .

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$$

- Substraemos el promedio de cada rostro.

$$\phi_i = \Gamma_i - \Psi$$

- Calculamos la matriz de covarianza C .

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T$$

$$A = [\phi_1 \phi_2 \dots \phi_M]$$

Donde: C es una matriz $N^2 \times N^2$ y A es una matriz $N^2 \times M$.

- Calculamos los autovectores μ_i de $C = AA^T$, con ayuda de los autovalores ν_i

$$\mu_i = A\nu_i$$

- Sólo conservamos los autovectores correspondientes a los "K" autovalores más altos.

b) Reconocimiento:

- Tenemos una imagen desconocida Γ con dimensiones de $[N^2 \times 1]$.
Normalizamos:

$$\phi = \Gamma - \Psi$$

- Proyectamos la imagen normalizada en el espacio característico.

$$\hat{\phi} = \sum_{i=1}^K w_i \mu_i$$

- Representamos como un vector Ω :

$$\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_K \end{bmatrix}$$

- Calculamos la distancia euclidiana.

$$d = \min \|\Omega_r - \Omega\|$$

- Nos quedamos con la clase que tenga la menor distancia.

2.4.4) Problemas del Eigenface

- Si una imagen es volteada no se reconocerá, esto quiere decir que tiene problemas a la hora de trasladar la imagen.
- El efecto zoom; o lo que es lo mismo, la variación en el tamaño del objeto produce errores en el cálculo dificultando su reconocimiento.
- Si el fondo de la imagen cambia, dificultará su reconocimiento.
- Si la iluminación cambia (intensidad de la luz) el rostro no podrá ser detectado.

2.5 Clasificadores en Cascada

Esta técnica de detección de rostros, presenta tres etapas.

- Cálculo de la imagen integral.
- Extracción de características por medio de Filtros Base Haar.
- Clasificación usando AdaBoost.

2.5.1) Cálculo de la imagen integral

La imagen integral permite extraer de forma rápida características a diferentes escalas ya que no se trabaja directamente con los valores de intensidad si no con una imagen acumulativa que se construye a partir de operaciones básicas

Es la suma de los pixeles contenido en un área delimitada (x, y) . Para el cálculo de la imagen integral se aplica de forma recursiva a cada pixel de la imagen.

Utilizamos la siguiente fórmula:

$$ii(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y')$$

Donde $ii(x, y)$ es la imagen integral y $i(x, y)$ la imagen original.

2.5.2) Extracción de características: Filtros Base Haar

La detección de objetos se basa en tener en cuenta ciertas características (features). Éstas características se usan para codificar la existencia de contraste entre regiones en una imagen.

La extracción de características es realizada aplicando a la imagen filtros con bases Haar. Estos filtros pueden ser calculados eficientemente sobre la imagen integral, son selectivos en la orientación espacial y frecuencia, y permiten ser modificados en escala y orientación.

De esta manera, un conjunto de características pueden ser usadas para codificar los contrastes encontrados en los rostros.

La propuesta original considera 3 tipos características:

- **two-rectangle feature:** Su valor se calcula con la diferencia entre la suma de los píxeles dentro de las dos regiones rectangulares.
- **three-rectangle feature:** Su valor se calcula con la suma de los píxeles dentro de las dos regiones rectangulares exteriores y la substracción de los píxeles de la región interior.
- **four-rectangle feature:** Su valor se calcula con la diferencia entre las diagonales de los pares de rectángulos.

El detector de objetos en OpenCV está basado en la propuesta de Viola y considera más tipos de características mejorado por Lienhart.

2.5.3) Clasificador: AdaBoost

Esta etapa dentro del algoritmo de detección se encarga de asignar un conjunto de características dado a una clase con la que se encuentra una mayor similitud, de acuerdo a un modelo inducido durante el entrenamiento.

Boosting: es un método de clasificación que combina varios clasificadores básicos para formar un único clasificador más complejo y preciso. La idea se basa en la afirmación de que varios clasificadores sencillos, cada uno de ellos con una precisión ligeramente superior a una clasificación aleatoria, pueden combinarse para formar un clasificador de mayor precisión, siempre y cuando se disponga de un número suficiente de muestras de entrenamiento. La aplicación de clasificadores en cascada ha permitido obtener buenos resultados.

Para aplicar la técnica de boosting primero se debe establecer un algoritmo de aprendizaje sencillo (clasificador débil o base), que será llamado repetidas veces para crear diversos clasificadores base. Para el entrenamiento de los clasificadores bases se emplean, en cada iteración, un subconjunto diferente de muestras de entrenamiento y una distribución de pesos diferente sobre las muestras de entrenamiento.

Finalmente, estos clasificadores base se combinan en un único clasificador que se espera sea mucho más preciso que cualquiera de los clasificadores base por separado.

En función de los clasificadores bases que se utilicen, las distribuciones que se empleen para entrenarlos y el modo de combinarlos, podrán crearse distintas clases del algoritmo genérico de boosting. El algoritmo de boosting empleado por Viola y Jones en su trabajo es conocido como AdaBoos.

2.6 Seguimiento de rostros en video

En los apartados anteriores hemos trabajado con imágenes estáticas, abordando los problemas de detección de rostros y detección de componentes faciales; sin embargo, existe una amplia variedad de aplicaciones que requieren el manejo de fuentes de video en el procesamiento de rostros.

El seguimiento es un proceso iterativo realimentado por las sucesivas imágenes (o frames) que constituyen la secuencia de video. En principio, un seguimiento preciso implica también aplicar un localizador de componentes faciales, como los que acabamos de ver en los apartados anteriores.

Se podría decir que el seguimiento de rostros, no es más que la relocalización continua del rostro y sus componentes faciales a lo largo del video, pero además del problema de la localización de las imágenes estáticas, existe una dimensión temporal que se debe tener en cuenta. Esto quiere decir que existe información adicional sobre las caras que están siendo seguidas, como por ejemplo su posición, orientación, forma, velocidad, etc.; que es la historia de cada instancia a lo largo del tiempo.

El último paso del seguimiento es la de relocalizar el rostro en el nuevo frame; obviamente, cuanto más acertada sea la predicción, más sencillo será el problema.

2.6.1) El problema del seguimiento

- **El movimiento del rostro;** tiene un carácter altamente impredecible, en condiciones normales, las variaciones de posición serán muy lentas y estarán acotadas espacialmente; sin embargo, existen movimientos que aunque su velocidad no sea muy alta, la aceleración puede ser grande. Un ejemplo gráfico de lo explicado anteriormente es la figura 13.



Figura 13: Movimientos de Rostro

- **Desenfoque por movimiento;** en el laboratorio es improbable que se produzca éste problema; sin embargo, en aplicaciones orientadas a uso público en donde se utilizan cámaras de bajo coste y las condiciones de iluminación pueden no ser óptimas, es común observar cierto desenfoque (figura 14).



Figura 14: Velocidad del Movimiento

- **Tiempo real;** la eficiencia computacional adquiere un papel primordial cuando se trata de manejar fuentes de video. Aunque la experimentación se realice con secuencias grabadas, las aplicaciones estarán hechas para el procesamiento en tiempo real; por tanto, se impondrán fuertes restricciones sobre la complejidad computacional.
- **Oclusión y desaparición;** es la situación en la que la cara deja de verse de forma total o parcial en una secuencia de video, la más común es la salida por uno de los extremos de la imagen. También puede ocurrir oclusiones de elementos externos, como la mano del propio individuo o el pelo (figura 15).



Figura 15: Oclusión del Rostro

- **Posición y orientación 3D;** la mayoría de técnicas de seguimiento suponen que el individuo mira de frente a la cámara y admiten ciertos márgenes de inclinación y giro lateral y vertical (figura 16), algunas técnicas producen resultados poco fiables y otras simplemente pierden el seguimiento.

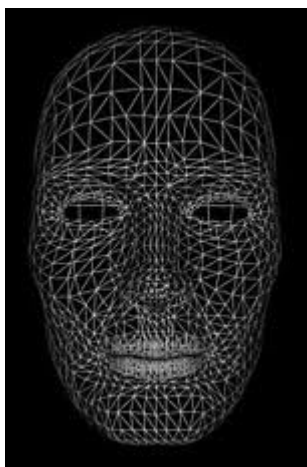


Figura 16: Rostro en 3D

- **Cambios de iluminación y expresión facial;** en relación a la iluminación puede ocurrir un cambio de matices de color, que dependiendo del lugar en el que será usado ésta aplicación podrá ser determinante o no, por ejemplo si se usa en plena vía pública o en el metro, las consideraciones son diferentes. Además se tendrá que tener en cuenta las expresiones faciales de la persona, que dificultará aun más la detección, véase figura 17.



Figura 17: Gestos en Rostro

2.6.2) Clasificación según el modelo de seguimiento

1. Seguimiento 2D:

Realizan un seguimiento plano del rostro, que incluye las estimaciones de posición y tamaño del rostro, es decir, tres grados de libertad. El resultado podría darse con un rectángulo que encierra el rostro.

2. Seguimiento 2,5D:

Además de la posición y el tamaño del rostro, se añade alguna información sobre orientación de la cabeza, típicamente la inclinación. Se pueden distinguir 2 clases, una sigue globalmente la cabeza y las otras siguen los elementos faciales, tales como los ojos y la boca, u otro elemento.

3. Seguimiento 3D:

Se resuelven seis grados de libertad correspondientes a posición y orientación de la cabeza.

4. Seguimiento basado en puntos característicos:

El resultado, es un conjunto disperso de puntos que corresponden con las localizaciones pre-definidas del rostro. El grado de libertad depende del número de puntos; algunos métodos son capaces de deducir la pose a partir de los puntos, otros incorporan el contorno de la cara.

5. Seguimiento con mallas y modelos deformables:

Alternativa que presenta un mayor grado de libertad, se basa en la definición de un modelo genérico de rostro mediante una malla densa de puntos.

2.6.3) Clasificación según la estrategia de búsqueda

1. Seguidores basados en movimiento:

En estos métodos, la posición de la cara se deduce a través de técnicas de flujo óptico y análisis de movimiento. La característica común de estos sistemas es que se apoyan únicamente en las diferencias de píxeles entre un frame del video y el anterior.

2. Seguidores basados en modelos:

El proceso de seguimiento se puede interpretar como un ajuste del modelo a la nueva imagen, esto es, buscar la posición, orientación y los posibles parámetros del modelo que mejor encajen con la cara actual. Se distinguen dos sub categorías:

- Los primeros tienen la ventaja de la flexibilidad; ya que pueden funcionar de forma fiable y precisa con cualquier individuo. Su inconveniente es que la inicialización es crítica, por ejemplo si la imagen inicial no es muy representativa.
- Los segundos requieren un entrenamiento, que puede ser genérico o específico para cada usuario.

3. Basados en apariencia y características:

Intentan aprovechar toda la información de las imágenes y del modelo mediante un análisis a través de la síntesis, también se usan técnicas de gradiente descendente. Los que están basados en las características tratan de aprovechar las propiedades invariantes y fáciles de seguir.

2.6.4) Mecanismos de predicción

Muchos algoritmos de seguimiento facial incorporan filtros avanzados de predicción, pero también existen bastantes propuestas que obvian el predictor basándose en la hipótesis de que la posición esperada en el nuevo frame corresponde a la observada en el instante anterior por tanto se trabaja con el pasado y no con el futuro.

1. Predicción mediante Filtros de Kalman:

El sistema es modelado con cuatro componentes: el vector de estado interno del proceso, la observación o medición obtenida, una variable de ruido de medición y el ruido en el proceso.

Existen muchas variantes del mecanismo original, una de las más interesantes son los llamados filtros de Kalman extendidos, que se aplican cuando las funciones no son lineales.

2. Predicción con el algoritmo Condensation:

Los filtros Kalman suponen una distribución gaussiana para la función de densidad de probabilidad del estado $p(X_t)$, lo cual puede resultar inadecuado para modelar la incertidumbre de ciertos procesos. Para solventar esta carencia, se propone el algoritmo condensation.

Este algoritmo permite mantener varias hipótesis simultáneas, dando lugar a una distribución multimodal; por tanto, el resultado no es una estimación concreta, sino una función de densidad de probabilidad.

La principal aportación de ésta técnica, es la capacidad de adaptación a situaciones donde el movimiento resulta impredecible, normalmente lento, pero rápido y espontáneo en ciertos tramos.

2.6.5) Seguimiento de caras basado en color y otras características

1. Algoritmo CamShift:

Es una extensión del algoritmo Mean Shift, el autor describe un interface perceptual completo, desde el proceso de seguimiento hasta la transformación de los resultados en variables de control del interface. Para modelar el color de piel se utiliza el histograma del canal Hue del espacio HSV, descartando los píxeles con bajo valor de saturación; también se eliminan los que sean muy claros o muy oscuros. El histograma se obtiene de la primera imagen de la secuencia y se aplica sobre las restantes, produciendo imágenes de probabilidad de color de piel.

El algoritmo parte de una ventana de búsqueda inicial (v) y lleva a cabo los siguientes pasos:

- Calcula la media de la imagen de probabilidad (p) dentro de la ventana (v).
- Mover el centro de la ventana (v) según el resultado del paso 1, y cambiar el tamaño según la suma de (p) dentro de (v); es decir, la ventana aumenta o disminuye según las probabilidades que contiene.
- Repetir los pasos anteriores hasta alcanzar la convergencia.

El método es conceptualmente sencillo, fácil de implementar, robusto, adaptable a movimientos rápidos y muy eficientes; no obstante, sus resultados son muy imprecisos, ya que la cabeza es descrita como una nube gaussiana en la que no se conoce la posición de los elementos faciales. Además la forma de la nube puede verse afectada por la aparición de distractores externos (mano, cuello, otras personas, etc.).

El algoritmo CamShift en sí es muy rápido, pero cuando trabajamos con imágenes grandes, la transformación RGB a HSV puede ser demasiado costosa.

2. Seguimiento mediante blobs:

LAFTER (lips and face tracker), desarrollado por Oliver, entre otros. Las principales diferencias con el algoritmo de camShift son:

- En lugar de usar HSV, se toman los canales R y G del espacio RGB.
- Los modelos de color se describen usando mezclas de gaussianas, cuyos parámetros son estimados con el algoritmo EM. Existe un modelo para el color de piel, otro para el fondo y otro para el conjunto boca/labios.
- La agrupación en regiones de color de piel se realiza con un algoritmo de clustering usando una representación de blobs para los conjuntos de píxeles; esto es, para cada píxel se forma un vector que combina posición y color.
- Los modelos de color se actualizan de forma continua a lo largo del video. Para evitar una oscilación imprevista, se utilizan filtros de kalman.

Un posible inconveniente relacionado con el seguimiento mediante color es el cambio de la iluminación en la escena.

2.6.6) Seguimiento de caras basado en apariencia

1. Algoritmo de alineamiento iterativo de Lucas Y kanade:

Dadas dos imágenes en instantes sucesivos, el objetivo es encontrar el desplazamiento que ha tenido lugar en un píxel dado (X,Y) que denotamos como (v_x, v_y) .

Evita la búsqueda exhaustiva y supone que el desplazamiento ocurrido es pequeño, tenemos que encontrar los valores de (v_x, v_y) que hagan cumplir la ecuación:

$$I_t(x, y) = I_{t+1}(x + v_x + y + v_y)$$

Idealmente, con una sola ejecución se obtendría el valor óptimo de (v_x, v_y) que genera el menor residuo. En la práctica se requiere varias iteraciones del proceso hasta alcanzar la convergencia, aunque por fortuna se consigue muy rápidamente.

Una limitación de éste algoritmo es que el máximo movimiento permitido debe ser pequeño, y nunca mayor que el tamaño de la ventana, que tampoco puede ser muy grande para cumplir la suposición de movimiento uniforme.

- **Tamaño de la ventanas;** permite definir una escala de posibilidades que va desde el seguimiento de pequeñas regiones características (fosas nasales, esquina de la boca, etc.), hasta la aplicación sobre la imagen completa de la cara.
- **Imagen de referencia;** en la practica el alineamiento se suele hacer respecto de un modelo, que puede ser una imagen de la misma secuencia o un modelo estadístico entrenado.
- **Funciones de modelado del movimiento;** cuando el tamaño de las ventanas es muy pequeño, el simple modelo de traslación es suficiente, pero a medida que el tamaño aumenta, se requieren funciones más complejas para describir el movimiento facial. Se han usado modelos de movimiento plano (rotación, traslación y escala, RTE, etc.).
- **Movimientos no rígidos e iluminación;** se aborda mediante la separación de forma y textura con modelos de malla deformables; otra forma es con la utilización de autoespacios. En los primeros se define un mapeo entre la imagen actual y un modelo rectificado, trabajando sobre las imágenes rectificadas. En los segundos se separan las distintas fuentes de variación con PCA o técnicas similares.

En definitiva, los métodos basados en apariencia son actualmente uno de los que ofrecen mayor precisión, detalle y estabilidad en el seguimiento 3D de las caras. Algunos de sus principales inconvenientes, en comparación con los métodos basados en características, son su elevado coste computacional, la dificultad de adaptación frente a variaciones rápidas, la re inicialización tras las pérdidas del seguimiento, y la mayor complejidad del entrenamiento de los modelos.

CAPITULO III

Visión Artificial



La visión es el tacto del espíritu.

Fernando Pessoa

3.1 Visión Artificial

Historia del visón artificial

A través de una cámara se obtienen imágenes planas de un mundo tridimensional. Pero el interés en representar este mundo tridimensional en dos dimensiones ya existía desde el mundo antiguo, como es el caso de la pintura.

Los siguientes grandes avances tuvieron lugar mucho tiempo después, ya en el Renacimiento. Los pintores italianos fueron los primeros en entender la formación de las imágenes y en estudiar la geometría con el objetivo de reproducir correctamente los efectos de la perspectiva, en contraposición con la pintura de épocas anteriores, que no mostraba diferencias de profundidad en los objetos. Como ejemplo, véase la siguiente imagen:



Figura 18: Iglesia Renacentista

En el año 1826, valiéndose de una cámara oscura y una superficie fotosensible, el químico francés Niepce (1765-1833) toma la primera fotografía. La metodología tuvo que ser perfeccionada y es en 1838 cuando otro químico de la misma nacionalidad, Daguerre (1787-1851) hizo el primer proceso fotográfico práctico, utilizando una placa fotográfica que era revelada con vapor de mercurio y fijada con trisulfato de sodio.

En la actualidad se utilizan cámaras réflex y CCD que emplean lentes para incrementar la potencia de la luz y mejorar el enfoque de la imagen.

Los comienzos

Se podría considerar que las raíces de lo que es en la actualidad el campo de la visión artificial están en el programa espacial de la NASA, en 1964. El programa espacial utilizaba, en un principio, cámaras de televisión que enviaban imágenes analógicas de vuelta a la Tierra.

Posteriormente, se cambió a la idea de usar una cámara digital y enviar la información en bits y bytes desde el satélite Mariner 4, en 1964, mientras volaba al planeta Marte. Fue con el procesamiento digital de dichas imágenes con lo que se dio inicio a lo que es en la actualidad el campo de la visión artificial.

La visión artificial es un campo de estudio diverso y relativamente nuevo. En los inicios de la computación era complicado procesar incluso conjuntos moderadamente grandes de datos de imagen. No fue hasta finales de los años setenta que emergió un estudio más concentrado de dicho campo.

La visión artificial cubre un ancho rango de temas que suelen estar relacionados con otras disciplinas y, por tanto, no hay una formulación concreta y estándar de lo que es "el problema de la visión artificial". Más incluso, no hay una formulación siquiera de cómo han de ser resueltos los problemas de visión artificial. En lugar de eso, existen abundantes métodos para resolver varias tareas bien definidas de visión artificial, en donde la metodología es generalmente muy específica y rara vez puede ser generalizada para un amplio rango de aplicaciones. Muchos de los métodos y aplicaciones están aún en un estado básico de investigación, pero varios de ellos han logrado hacerse un hueco como productos comerciales, a menudo formando parte de un sistema más grande capaz de resolver problemas complejos.

En la mayor parte de las aplicaciones prácticas que usan visión artificial esta es pre-programada para resolver una tarea en particular, pero los métodos basados en aprendizaje se están volviendo cada vez más comunes.

La visión artificial abarca gran variedad de campos como la informática, la óptica, la ingeniería mecánica, la ingeniería industrial, la robótica, etc. Las aplicaciones de visión artificial industrial integran sistemas de captura de imágenes digitales, dispositivos de entrada/salida y redes de ordenador para el control de equipos destinados a la fabricación, tales como brazos robóticos.

Los sistemas de visión artificial se destinan a realizar inspecciones visuales que requieren alta velocidad, funcionamiento las 24 horas del día o la repetitividad de las medidas.

Cómo funciona la visión por computador

Los pasos básicos son la segmentación y el análisis, que son esencialmente algoritmos de software y funciones que se ejecutan en el procesador.

1) Segmentación

Se trata de decidir qué partes de la imagen necesitan interpretación o análisis y cuáles no.

A menudo es posible y necesario refinar la segmentación. Por ejemplo, si la aplicación debe encontrar arañazos o grietas en un vidrio, la primera fase de segmentación suele encontrar elementos que son las propias grietas o arañazos además de suciedad y polvo. A menudo se perfecciona este resultado para eliminar de la imagen la suciedad y el polvo de los elementos a analizar.

2) Análisis

Una vez que la imagen ha sido satisfactoriamente segmentada, el procesador puede hacer una serie de pruebas y mediciones en los elementos de interés que aparecen en la escena. Existen multitud de algoritmos que nos ayudan a obtener los resultados deseados.

Entre ellos, cabe destacar sus funcionalidades:

- Detección de formas.
- Análisis geométrico.
- Comparación con patrones.
- Medición de objetos.
- Detección de objetos difusos.

Estos algoritmos, usados estratégicamente, nos permiten decidir si una pieza se ha fabricado correctamente o no, guiar un robot hacia un objeto concreto o reajustar la maquinaria de producción automáticamente.

3.1.1) Porque el uso de la visión artificial

Las inspecciones realizadas por los seres humanos, a menudo no pueden cumplir con los requisitos de la industria moderna respecto a la velocidad de producción, calidad de producto y costes de producción. Los humanos se cansan, cometen errores y los criterios que se aplican durante las inspecciones son inevitablemente subjetivos. En algunos casos, no es humanamente posible llevar a cabo las tareas de inspección debido a las condiciones ambientales. Las cámaras y los sistemas que componen un sistema de visión artificial, por el contrario, llevan a cabo las mediciones con una precisión constante y a un ritmo que es establecido por el propio proceso de producción. Estas ventajas han llevado a una creciente aceptación de la visión por industrias de todo el mundo, con la esperanza de que un día, no muy lejano, los robots puedan reemplazar a los humanos, para hacer tareas peligrosas como por ejemplo trabajar en condiciones de temperatura extrema.

La visión por ordenador ha otorgado a la robótica una gran solución para proporcionar a sus robots un aspecto humanoide, como vemos en la siguiente imagen (Figura 19), donde se aprecia a un robot humanoide y su campo visual.

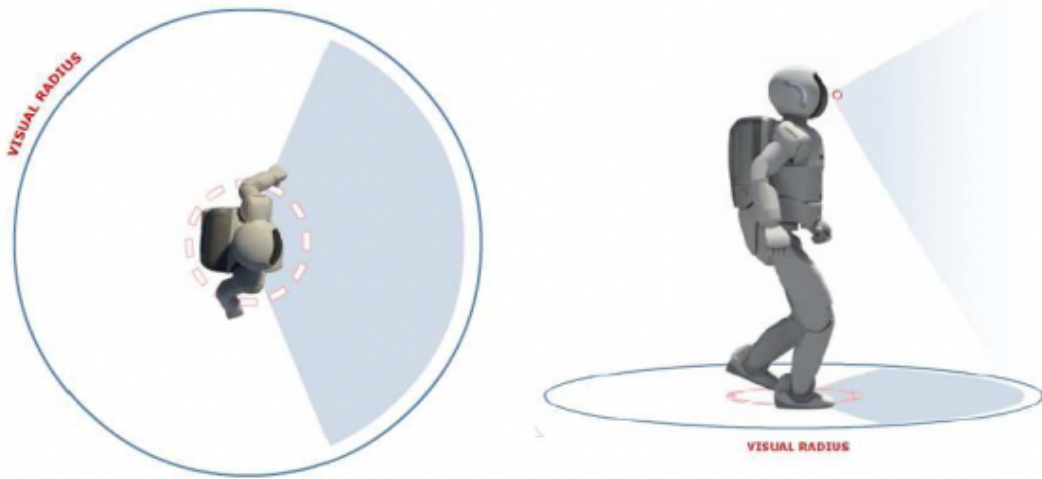


Figura 19: Alcance de la visión de un Robot

Otro de los objetivos es diseñar herramientas tecnológicas que permitan a las personas la mejora de su calidad de vida, por ejemplo creando aplicaciones para mejorar la seguridad en el tráfico, o robots para la limpieza del hogar, etc.

Aplicaciones de la visión por computador

Con el incremento de potencia de los microprocesadores, las aplicaciones que pueden resolverse con éxito utilizando sistemas de visión están creciendo rápidamente. Una aplicación puede exigir una o más funciones de procesamiento de imágenes que, cuando se combinan, crean una solución. La gama de detección es muy amplia e incluye:

1) Forma o apariencia: Control de la conformidad

Los sistemas actuales, por lo general, comienzan con una operación de medición bidimensional (Figura 20) para establecer el desplazamiento necesario para que el objeto se encuentre en una posición ideal. Posteriormente, se llevan a cabo las operaciones de comparación con un objeto patrón o la caracterización geométrica de su forma. De esta manera, se puede decidir si el objeto analizado cumple o no con los requisitos preestablecidos.



Figura 20: Medición de piezas

2) Detección de defectos: Elementos discretos

La detección de defectos puede ser considerada como un caso especial de control de la conformidad, en el que los objetos no ofrecen formas específicas que lo caracterizan. Como ejemplos, citaremos la inspección de baldosas, cerámicas, en la etapa previa al esmaltado para detectar defectos en la superficie como rugosidades, o el uso de infrarrojos para la inspección de defectos de botellas de vidrio.

3) Detección de defectos: Tejidos o laminados

La característica de muchos sistemas de este tipo es que se debe examinar grandes áreas a gran velocidad, en busca de defectos relativamente pequeños. El material producido en bobinas incluye la chapa de acero, papel, películas fotográficas, vidrio, hoja de plástico y los textiles. Algunos de estos sistemas utilizan el escaneado láser en lugar de cámaras lineales para formar la imagen.

4) Control por colores

La inspección por color se utiliza ampliamente en la industria farmacéutica para confirmar que la tableta o píldora de color se ha colocado en el embalaje correcto, en la gráfica siguiente (Figura 21) veremos un ejemplo de esta técnica.



Figura 21: Control por Colores

5) Medidas unidimensionales o bidimensionales

Las mediciones bidimensionales se pueden realizar mediante la superposición de calibres ópticos sobre la imagen del producto. La medición por este método es muy rápida porque no hay movimiento mecánico, y el posicionamiento exacto del objeto a medir no es crítico (el sistema puede determinar el desplazamiento X e Y del objeto y el grado de rotación).



Figura 22: Medidas Bidimensionales

Algunos sistemas ofrecen interpolación sub-píxel que, en condiciones adecuadas, hace que el sistema sea capaz de medir y, en particular, detectar cambios o diferencias en la medición, de milésimas en lugar de las centésimas que ofrecen otros métodos de metrología. El número de píxeles de la imagen se puede aumentar mediante el uso de cámaras de muy alta resolución y obtener así mediciones aún más precisas.

6) Luz estructurada y otras técnicas de triangulación

Si una banda fina de luz se dirige a una superficie tridimensional y se observa con una cámara en un ángulo diferente al del dispositivo de iluminación, la

forma aparente de la banda puede ser usada para inferir la forma de la superficie expuesta a lo largo de la longitud de la banda.

Mediante la exploración de la banda a través de la superficie, se puede generar un mapa tridimensional del objeto observado. A menudo es conveniente usar un láser como fuente de luz, dado que puede ser fácilmente refractado para formar un plano de luz, como se muestra en la figura.



Figura 23: Luz estructurada

Técnicas tridimensionales

También se puede obtener información tridimensional mediante el uso de visión estereoscópica automatizada.

1) Reconocimiento de caracteres

El reconocimiento de caracteres se puede basar en técnicas de correlación simple (comparación), sin conocimientos especializados de la forma en que los caracteres individuales se forman, o se pueden usar algoritmos más avanzados de inteligencia artificial.

Los sistemas de visión también pueden ser usados para leer códigos de barras con un mayor grado de confianza que un escáner láser. Incluso si una gran parte del código de barras está oculto por suciedad, un sistema de visión, por lo general, puede leerlo, mientras que un lector de códigos de barras convencional no.

Además de códigos de barras, los sistemas de visión también se usan para leer códigos matriciales de dos dimensiones, donde los códigos pueden contener mucha más información que un código de barras unidimensional.



Figura 24: Código de Barras

2) Reconocimiento de piezas o componentes del producto

El reconocimiento de piezas o componentes puede ser usado para identificar qué partes deben ser sometidas a cada proceso. Por ejemplo, un sistema de visión puede discernir entre las diferentes piezas de un elemento e instruir a un robot para el correcto montaje del elemento.

a) Guiado predeterminado

El guiado predeterminado (Figura 25) se caracteriza por una situación en la que una cámara azimutal toma una instantánea de la escena y el sistema de visión dirige un robot para que recoja o deje un objeto en una posición concreta. A continuación, el robot obra a ciegas. Una aplicación típica incluye tareas de palatización de objetos pesados, tales como bloques de motor y cigüeñales, pero también puede incluir trabajos mucho más ligeros, tales como el embalaje de chocolates.



Figura 25: Guiado predeterminado

b) Guiado continuo

El guiado continuo implica una cámara montada (Figura 26) sobre el brazo o la mano del robot y el camino del robot está continuamente corregido por el sistema de visión. Una aplicación común es el guiado de la soldadura por arco, pero la técnica puede utilizarse también para controlar el camino en la aplicación de selladoras o pegamentos por robots.



Figura 26: Guiado continuo

3.1.2) Componentes de un sistema de visión artificial

Estos componentes pueden variar, dependiendo del uso que se le dará a la visión por ordenador. Aquí les presentaremos los componentes necesarios para obtener una buena visión artificial en el sector industrial.

- **Un sistema de iluminación.**

Una buena iluminación (ver figura 27) es especialmente importante para la toma de imágenes de los productos en una línea rápida de producción, aunque algunas aplicaciones pueden utilizar la luz ambiente.



Figura 27: Fuentes de Luz

- **La lente de la cámara.**

La correcta selección de lentes es importante para alcanzar una solución óptima, a continuación en la figura 28 mostramos una gran variedad de lentes.



Figura 28: Lentes

- **Una o más cámaras para adquirir las imágenes.**

El precio de las cámaras digitales (ver figura 29) está disminuyendo, de modo que estas se están usando cada vez en más número; ya que, mientras más perspectivas tengas mejor, además es la base para obtener imágenes en 3D.



Figura 29: Cámaras

- **Un dispositivo de interfaz para transferir las imágenes al ordenador.**

- **Un procesador de imagen, ordenador o cámara inteligente.**

Una opción es utilizar cámaras inteligentes que integran el procesamiento de imágenes dentro de la propia cámara, evitando la necesidad de transferir imágenes a un ordenador externo(ver figura 30).

La velocidad de proceso de estas cámaras es inferior a la de un ordenador y existen aplicaciones en las que estas no son adecuadas.



Figura 30: Ordenador

- **Una interfaz para notificar el resultado del análisis a un operador.**

Se puede notificar de este resultado mediante una señal electrónica que opera un mecanismo de rechazo.

La imagen de entrada; una matriz bidimensional de niveles energéticos (por ejemplo, luz), se divide en elementos de imagen, conocidos como píxeles. Estos forman filas y columnas que abarcan toda la zona de la imagen y representan los niveles de gris en una imagen monocromática o la codificación de color en una imagen en color. Un píxel no puede ser subdividido en regiones de menor nivel de gris o color. Este proceso es un tipo de digitalización espacial. Para cada píxel, la información del nivel de energía también deben ser digitalizado, es decir, los niveles analógicos (variable continua) producidos por la cámara deben ser representados por un número finito de pasos. En muchas aplicaciones es suficiente digitalizar una imagen monocroma con 8 bits por píxel, lo que equivale a 256 pasos, para representar el nivel de gris de cada píxel. En aplicaciones más exigentes puede ser necesario digitalizar a 14 bits (o 16384 niveles). Las imágenes en color son más complejas y pueden ser representados en diferentes formatos. Las imágenes en color normalmente contienen tres veces más información que una imagen monocromática.

Algunos sistemas de visión no utilizan una cámara matricial, en su lugar se usa una cámara lineal que produce una sola línea o fila de píxeles. La imagen bidimensional se genera a medida que el objeto pasa bajo la cámara lineal, aprovechando su movimiento, normalmente generado por una cinta transportadora. Uniendo las distintas filas de píxeles obtenidas a diferentes intervalos de paso, se obtiene una imagen bidimensional.

3.1.3) En la Actualidad

A continuación les presentaremos unos cuantos artículos; uno de ellos publicado en el diario El país [14], en donde se pueden apreciar las últimas novedades sobre la visión de ordenador.

1) Avances en la agricultura:

El primer robot con visión artificial para recoger la fresa automáticamente se ha presentado en sociedad en Huelva. La empresa onubense Agrobot ha probado ante posibles clientes su cosechadora, producto del trabajo de un grupo de ingenieros desde hace casi un año.



Figura 31: Robot recogedor de fresas

El desarrollo del robot lo ha dirigido el ingeniero Juan Bravo, quien ha conseguido plasmarlo en un prototipo que tiene unos 40 brazos robóticos y que identifica los frutos maduros y los recoge sin dañarlos. El ejercicio de prueba se ha realizado en la parcela experimental del centro tecnológico Adesva de la Junta de Andalucía, situado en Lepe (Huelva), y ha servido como primera toma de contacto de los agricultores, cooperativas y la patronal Freshuelva con este proyecto.

Según Bravo, además de recoger las fresas, esta cosechadora es capaz de clasificarlas y colocarlas en los envases que recorren las cintas transportadoras, y todo porque "el sector está sometido a demasiadas variables y consideramos que la maquinaria aumentará los beneficios gracias al aumento de la calidad de los frutos recogidos, que se hace de forma higiénica y sin presionarlos".

2) Un prototipo de visión artificial permitirá a ciegos sortear los obstáculos:

La Universidad Politécnica de Valencia ha coordinado el desarrollo de sendos prototipos con metodologías basadas en visión artificial, análisis acústico, sensores de información 2D y sistemas de posicionamiento GPS que permitirán a los ciegos sortear obstáculos y moverse en espacios abiertos y cerrados.

Concretamente se han creado dos prototipos, uno de ellos unas gafas que emiten un rayo láser que permite localizar objetos por sonidos a una distancia de cinco metros, y el otro, con un alcance de quince metros, basado en un sistema de estereovisión con dos cámaras ubicadas en un casco.

Guillermo Peris a señalado que el proyecto CASLiP "nació con el objetivo de diseñar una herramienta tecnológica que permita a las personas invidentes la integración en la vida social mejorando su calidad de vida y aportándoles herramientas adicionales a las que ya poseen, como son los bastones y perros".

Además, ha recordado que: "El 1,9 por ciento de la población europea tiene discapacidad visual, total y parcial, motivo por el cual la Politécnica de Valencia y sus socios decidieron abordar la investigación, mezclando para ello metodologías basadas en visión artificial, análisis acústicos y sistemas GPS (figura 32). Por su parte, el investigador de la Universidad de La Laguna José Luis González a comparado estos sistemas de visión artificial con un GPS utilizado en la actualidad, y con él la persona ciega podrá determinar el lugar al que desea, pudiendo sortear todos los obstáculos que se presente en su camino a través de los distintos tipos de sonidos que recibe".

Por último, ha precisado que se trata de dos prototipos "con los que se están trabajando aún, por lo que su comercialización podría llevarse a cabo dentro de tres o cuatro años".



Figura 32: Ceguera

3) Visión artificial al guiado de robots industriales en Rayco

“Este proyecto ha sido realizado el 16 de febrero del 2010 y ha supuesto el desarrollo de un sistema de visión para guiar un robot industrial en condiciones muy complejas”, señala Eusebio de la Fuente, profesor del Departamento de Ingeniería de Sistemas y Automática de la Universidad de Valladolid. En este caso, el robot debe hacer un seguimiento de distintos modelos de radiador es que se transportan colgados por la cadena de fabricación y adherir pegatinas sobre cada una de las entradas y salidas.

“El fin es tapar estos agujeros porque cuando los radiadores pasan a la cabina la pintura entra por ahí y luego no cierran bien”, detalla el experto, quien afirma que realizar este trabajo que lleva a cabo un operario es complicado si no se dispone de un robot con sistema de visión que se posicione en los agujeros y dirija automáticamente la pegatina.

En la imagen (Figura 33) se muestra el trabajo conjunto que realiza el brazo mecánico y la cámara, que le proporciona un sistema de visión.



Figura 33: Brazo mecánico

4) Proyecto de la universidad Carlos III de Madrid

16 noviembre de 2010.

“Han presentado un videojuego que se manejaba simplemente con el movimiento de las manos, como si se estuviera agarrando un volante virtual”.

“Estos nuevos sensores ofrecen información de profundidad, lo que resulta muy interesante cuando trabajas en sistemas de visión artificial”.

“En medicina, por ejemplo, con este tipo de sensores se puede crear un sistema automático de rehabilitación que guíe al paciente en sus ejercicios de rehabilitación sin tener que moverse de casa”

Científicos de la Universidad Carlos III de Madrid (UC3M) han desarrollado un sistema de comunicación basado en el movimiento y posición de las manos para el control virtual de un videojuego mediante una cámara de tiempo de vuelo e investigan las aplicaciones de este sensor en medicina, biometría, deporte o computación emocional.

Los investigadores del Grupo de Inteligencia Artificial Aplicada (GIAA) del Campus de Colmenarejo de la UC3M han presentado la aplicación en el último Salón Internacional de Material Eléctrico y Electrónico celebrado recientemente en Madrid. Los asistentes al stand de la empresa Infaimon, que ha colaborado en el proyecto, tuvieron la oportunidad de probar este interface con un video juego que se manejaba simplemente con el movimiento de las manos, como si se estuviera agarrando un volante virtual. Para ello, los científicos han empleado una cámara de tiempo de vuelo o TOF (Time-of-flight) con la que capturan en 3D los movimientos del usuario para después poder transmitirlos a un ordenador, que los procesa y transmite al coche del juego. “Lo más complejo ha sido conocer las características de la cámara para poder optimizar el movimiento y su integración con muchas aplicaciones”, comenta uno de los investigadores del GIAA de la UC3M, Daniel Sánchez, que ha realizado su proyecto de fin de carrera en el marco de esta investigación.

La gran ventaja de este tipo de cámaras es que aportan información tridimensional sin recurrir a los clásicos sistemas estereoscópicos de dos lentes. “Estos nuevos sensores ofrecen información de profundidad, lo que resulta muy interesante cuando trabajas en sistemas de visión artificial”, comenta Miguel Ángel Patricio, que coordina esta investigación en el seno del Departamento de Informática de la UC3M. El funcionamiento de una cámara TOF es relativamente simple: un anillo de infrarrojos emite luz que rebota en el cuerpo que se graba y vuelve a un sensor. Por el tiempo que transcurre en ese proceso, se puede calcular la distancia a la que se encuentran los diferentes objetos. “Nuestra idea - apunta Patricio - es poder aplicar este sensor a distintos problemas en los que actualmente trabajamos, como sistemas de video vigilancia, de identificación biométrica de rostros, el análisis del rendimiento deportivo del movimiento de los jugadores e interfaces hombre-máquina”, resume.

Múltiples aplicaciones

Estos investigadores, que trabajan en el Campus de Colmenarejo de la UC3M, centran ahora sus esfuerzos en analizar la información que se obtiene empleando este tipo de sensores. “Estoy convencido de que su uso revolucionará los sistemas de visión artificial en el futuro, porque los datos que se obtienen son mucho más ricos que los conseguidos mediante otro tipo de sensores tradicionales”, afirma el profesor, que apunta que solo hay que esperar a que la economía de mercado haga que el precio de los mismos disminuya, ya que ahora rondan los 6.000 euros por unidad, aproximadamente.

El reto que se plantean estos científicos en la actualidad es conseguir aplicar el potencial de estas cámaras en determinados campos. En medicina, por ejemplo, con este tipo de sensores se puede crear un sistema automático de rehabilitación que guíe al paciente en sus ejercicios de rehabilitación sin tener que moverse de casa. Estos investigadores también colaboran con el INEF en el desarrollo de unos criterios de análisis de la obesidad infantil mediante un sensor TOF, algo que hasta ahora se realiza mediante láser. Y las aplicaciones también alcanzan a la llamada “computación afectiva”, a través del desarrollo de aplicaciones HCI (Human-Computer Interface) que intentan examinar el estado de ánimo de una persona mediante la aplicación de algoritmos que analizan la información que proviene de una de estas cámaras tridimensionales.

5) Una apuesta por la visión artificial en el sector alimentario

21 de septiembre de 2010

La tecnología de rayos X se muestra como una novedosa tecnología de visión artificial que por sus características complementa las inspecciones realizadas en otras porciones del espectro, al permitir la inspección en el interior de los productos y de los envases. Multiscan Technologies ha desarrollado las gamas X10H y X20V para la inspección tanto de productos envasados como de artículos a granel, con modelos de aplicación en una amplia gama de productos agroalimentarios.

Esta maquinaria tiene como principal misión comprobar la posible presencia de elementos contaminantes en el producto, pero además realiza la evaluación de distintos parámetros de calidad en la misma inspección, determinando desde la adecuación de los sistemas de cierre hasta la comprobación en el interior de los envases cantidades, pesos, niveles, etc.

Es posible detectar metales, metales no férricos, acero inoxidable, vidrio, piedras, huesos calcificados, PVC, teflón y materiales cerámicos en envase metálico, en envase de vidrio, en bolsas, bandejas, o en producto a granel.

Gracias al particular desarrollo de su software, las máquinas Multiscan trabajan a gran velocidad, cuidando específicamente los siguientes aspectos: la clara detección de defectos, la reducción de falsos rechazos, el control de la expulsión y eventos, y la trazabilidad del producto rechazado.



Figura 34: Máquina multiscan

6) Visión artificial para que no se escape ni un hilo

La empresa almeriense Emevision ha aplicado un sistema de visión artificial a la fabricación de mallas de plástico para invernaderos. El sistema comprueba que el tejido resultante esté completo y no falten hilos que den lugar a agujeros por los que se colarían insectos.

Manuel Luis Pérez dice lo siguiente al respecto:

El ojo humano es una máquina compleja de la naturaleza, pero no es perfecta. Como si de ciencia ficción se tratara, la informática ha desarrollado sistemas de visión artificial que, aunque no están conectados a un cerebro humano que interprete sensaciones en las imágenes que recibe, puede aplicarse en la detección de imperfecciones en cadenas de producción.

La empresa Emevisión, al frente de la que se encuentra José David García, lleva tres años desarrollando sistemas de visión artificial aplicados, principalmente, al sector agroalimentario almeriense. Dicho así es difícil imaginar qué hacen unos ojos robóticos oteando entre frutas y verduras. Estos ojos no son más que unas cámaras que captan imágenes de las frutas que pasan por la cinta transportadora, y tras enviarlas a una pantalla de ordenador, un programa informático analiza el tamaño, color, diámetro y los defectos superficiales, para clasificarlas y descartar las no válidas o dirigir las a embalajes distintos según su grado de madurez.

En la figura 35, vemos una imagen de una malla a la que le falta un hilo, detectado por visión artificial/EMEVISION

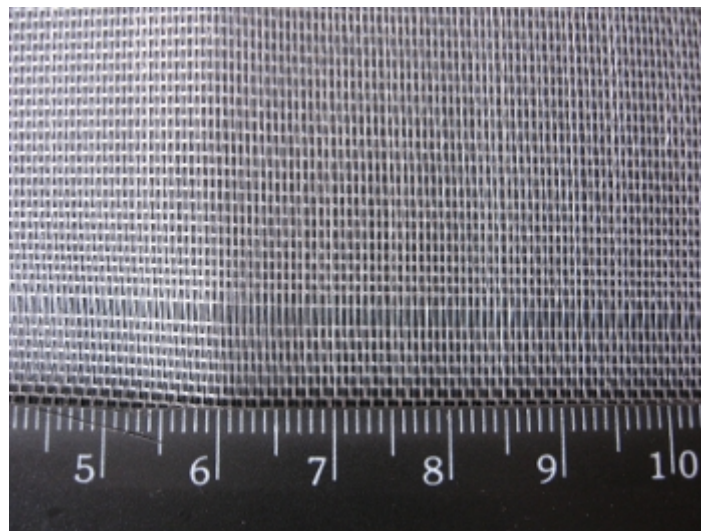


Figura 35: Aplicación a mallas

Los sistemas de visión artificial que desarrolla Emevisión son personalizados en función de las demandas de cada empresa. Así, en la actualidad está trabajando en la implantación de un sistema de visión artificial para controlar la fabricación de mallas de plástico para los invernaderos, que se colocan tanto en el exterior de los invernaderos como en las calles internas.

El software a medida de Emevisión detecta si en el tejido de la malla se produce algún desperfecto, como la falta de uno de los hilos. El sistema de visión artificial cuenta los hilos y la distancia entre los hilos. Así se evita que la malla presente agujeros por los que podrían entrar insectos perjudiciales para los cultivos. La cámara se coloca en el rodillo por el que va saliendo la malla y va tomando imágenes digitales para que el ordenador las analice.

El objetivo de la empresa es elaborar un programa informático de fácil manejo, de forma que un operario pueda controlar cómodamente el proceso sin necesidad de conocimientos avanzados. Toda la información va apareciendo en la pantalla del ordenador, avisando al operario con sirenas o sistemas oportunos.

Automatización

La visión artificial permite avanzar en los procesos de automatización de las fábricas, de forma que se consigue mayor rapidez en la cadena de producción y se evita exponer a los trabajadores a ambientes peligrosos o incómodos. El gerente, José David García, pone como ejemplo los sistemas aplicados en dos líneas de producción de una conocida empresa fabricante de sanitarios, donde han conseguido hacer una descarga más rápida de las piezas gracias a la rápida identificación proporcionada por la visión artificial.

3.1.4) Justificación de la inversión en visión artificial

Hay muchas justificaciones para la utilización de la visión artificial:

Costes de los materiales:

En la mayoría de las aplicaciones, evitar la producción de piezas defectuosas mediante el uso de un sistema de visión industrial tendrá un período de amortización muy corto. Para evitar que se fabriquen piezas defectuosas, el sistema de inspección automática, ya sea muestreando el 100% en la línea de producción o bien usado fuera de línea tomando muestras, debe formar parte del control estadístico de procesos del sistema productivo. Esto significa que el sistema indica cuando un parámetro de control deriva hacia el límite de tolerancia, o es simplemente demasiado errático. El sistema de visión puede tomar medidas correctivas antes de que el límite sea superado.

Costes de la mano de obra:

La reducción de la mano de obra es también un importante ahorro de costes, ya que muchas de las tareas realizadas por la visión industrial pueden sustituir a personas directamente. Además, deben ser considerados los ahorros en selección de personal, prestaciones sociales y los aumentos salariales anuales.

Costes de la calidad:

La creciente conciencia del costo de la calidad a través de las normas, significa que el uso de la visión artificial puede ofrecer un estándar más objetivo, fiable y consistente en la inspección de productos.

El ahorro en la optimización del uso de materiales, seguimiento de la calidad de los proveedores y garantía de calidad de los productos acabados pueden llevar a ahorros tanto tangibles como intangibles. El costo de los trabajos de reparación en garantía se puede reducir y además se mejora la confianza de los clientes consiguiendo pedidos recurrentes y una mayor cuota de mercado.

Ventajas de la Visión Artificial para el control de calidad

La Visión Artificial es una herramienta muy potente para el control de calidad en la industria. La gran versatilidad, elevada cadencia de inspecciones y escrupulosa precisión, hacen que la Visión Artificial haga posible la inspección de la totalidad de la producción.

- Inspección del total de la producción

Debido a la elevada cadencia que puede alcanzar la inspección con cámaras, la Visión Artificial para el control de calidad se puede usar para inspeccionar cada una de las piezas producidas en lugar de hacer muestreos aleatorios de los lotes.

- Repetitividad

Con la visión industrial para el control de calidad, cada una de las inspecciones es realizada bajo las mismas condiciones, asegurando así la repetitividad de cada uno de los parámetros de aceptación del producto.

- Consistencia

La inspección automatizada evita la subjetividad característica del ser humano y el cansancio, por lo que cada una de las piezas es consistente a nivel de calidad con todas las demás.

- Reducción de costes

Son evidentes las ventajas del control de calidad al final del proceso productivo respecto a la reducción de costes, servirá a su cliente productos libres de defectos, evitando devolución de lotes y mejorando su imagen. Con la inspección de los parámetros de calidad con cámara en puntos intermedios en su proceso productivo, le permitirá detectar con antelación piezas defectuosas y retirarlas antes de ser ensambladas, evitando así el rechazo completo del producto final. También obtendrá beneficios debido a la recolocación de recursos humanos previamente destinados a asegurar la calidad del producto.

3.2 Concepto de las OpenCV

En este y en los próximos apartados daremos una explicación sobre las librerías de las OpenCV y algunas de sus funciones. Veremos las principales características y como nos ayuda en el desarrollo del seguimiento de rostros. También hablaremos de sus limitaciones y posibles competidores.

A continuación mostramos el logotipo característico de las OpenCV:



Figura 36: OpenCV

¿Qué son las OpenCV?

OpenCV significa Open Source Computer Vision Library; por lo tanto, es una librería de tratamiento de imágenes, destinada principalmente a aplicaciones de visión por computador en tiempo real. Una de las ventajas principales es que puede funcionar en muchas plataformas, existen versiones para Windows, Linux y MacOs.

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel; desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

OpenCV es multiplataforma, Existiendo versiones para GNU/Linux, Mac OSX y Windows. 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.

¿Por qué usar las OpenCV?

El proyecto pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado, realizando su programación en código C++ optimizados, aprovechando además las capacidades que proveen los procesadores multi-núcleo. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel.

Las OpenCV además, cuentan a su disposición con dos funciones que tienen gran importancia para el correcto funcionamiento del proyecto; ya que, son la base para la detección de rostros, y son las siguientes:

- **CvHaarClassifierCascade* cvLoadHaarClassifierCascade(const char* directory, CvSize origWindowSize)**

Carga un clasificador en cascada previamente entrenado desde un archivo

directory: ejemplo: "C:/OpenCV/data/haarcascades/haarcascade_frontalface_default.xml"
origWindowSize: tamaño del objeto de la cascada.

- **CvEq* cvHaarDetectObjects(Img, cascade, storage, scale_factor, int min_neighbors=3, int flags=0)**

Detecta objetos en la imagen

Img: imagen donde se detecta el objeto.
cascade: representación del clasificador en cascada.
storage: Memoria reservada para almacenar el resultado.
scale_factor: factor de escala con la cual buscará la ventana.
min_neighbors: mínimo número de rectángulos vecinos.
flags: usa normalmente CV_HAAR_DO_CANNY_PRUNING.

La función cvHaarDetectObjects, es muy costoso computacionalmente hablando, y es por eso que sólo lo usamos para detectar el rostro la primera vez y luego para hacer refrescos cada cierto tiempo. Así pues, el proyecto consiste en optimizar esta función e interactuar con él mediante una estimación de movimiento, ayudándole en la detección en casos en el que éste falla y en aumentar su rapidez.

Las OpenCV Presenta 5 librerías como estructura fundamental.

- **cxcore** : estructuras básicas
- **cv** : principales algoritmos
- **cvaux** : algoritmos más experimentales
- **higui** : GUI and Video I/O
- **ml** : machine learning

Para más información sobre las OpenCV y con similares tecnologías, consultar con los textos descritos en la bibliografía: [1], [2], [8], [11], [12] y [13].

3.2.1) Representación Gráfica

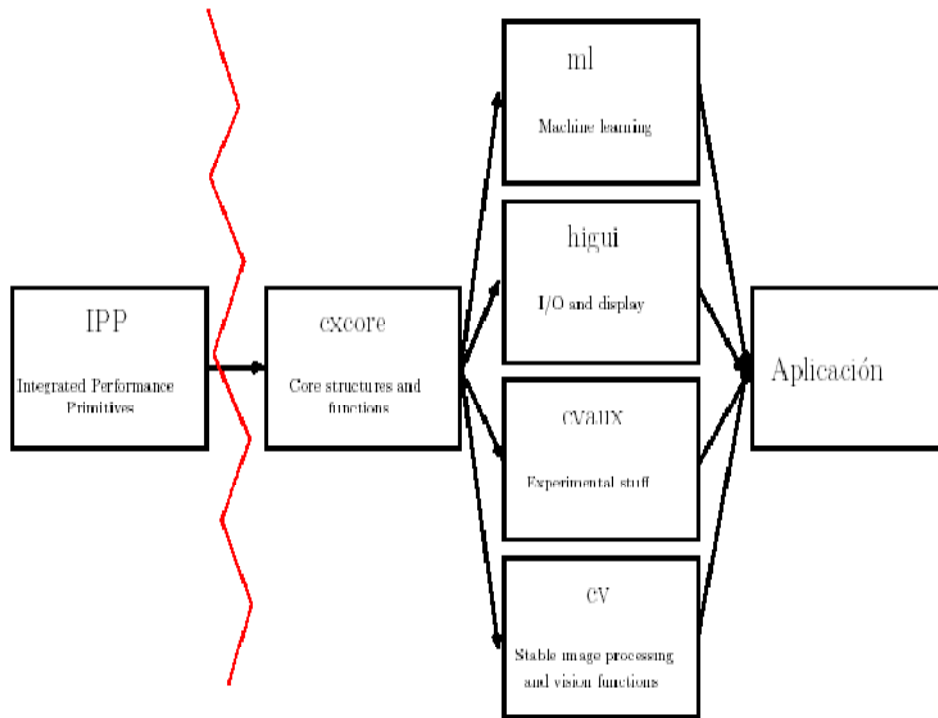


Figura 37: Representación OpenCV

Inconvenientes de las OpenCV:

Dadas las grandes posibilidades que ofrece OpenCV para el tratamiento de imágenes, calibración de cámaras, y otras muchas aplicaciones más como por ejemplo, para simular una prótesis ocular basada en un implante cortical y estudiar el funcionamiento de las retinas artificiales, etc.

Quizá de los pocos inconvenientes que se pueden encontrar en ella sea en el caso del seguimiento de objetos, en el cual, el principal inconveniente es que no ofrece un producto completo, tan sólo algunas piezas que sirven como base para montar sobre ellas un producto final.

Otro de los inconvenientes que tiene es la necesidad de utilizar la librería IPL para tener acceso a funciones de bajo nivel.

Sin embargo, la presencia de funciones muy interesantes, y las posibilidades ya comentadas que ofrece la librería hacen que estos inconvenientes no sean realmente significantes.

3.2.2) La Competencia

Además de las OpenCV, existen otras estrategias para controlar el procesamiento de imágenes y la visión por ordenador, mirar [11], [12] y [13].

- **LTI-Lib**:

Es una biblioteca orientada a objetos con algoritmos y estructuras de datos utilizadas en el procesamiento de imágenes y visión por ordenador. Fue desarrollado en la Universidad RWTH-Aachen, como parte de muchos proyectos de investigación en visión por computador, se ocupan de la robótica, reconocimiento de objetos, el lenguaje de señas, y reconocimiento de gestos. Proporciona un C++ orientado a objetos que incluye la colección de algoritmos rápidos, que pueden ser utilizados en aplicaciones reales. El símbolo de esta librería se aprecia en la figura 38.



Figura 38: LTI

- **VXL**:

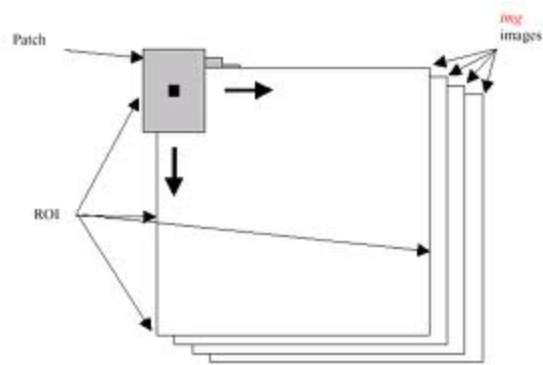
Es una colección de librerías C++, diseñado para la investigación en visión por ordenador y su aplicación. Fue creado a partir de TargetJr y el IUE, con el objetivo de hacer un sistema rápido y consistente. VXL está escrito en ANSI / ISO C++ y está diseñado para ser portátil en muchas plataformas. Además de estas librerías (figura 39), hay otras que cubren algoritmos numéricos, procesamiento de imágenes, la geometría de la cámara, manipulación de video, recuperación de estructuras en movimiento, diseño de interfaces gráficas de usuarios, imágenes 3D y mucho más.



Figura 39: VXL

CAPITULO IV

Seguimiento



La mayoría de las ideas fundamentales de la ciencia son esencialmente sencillas y, por regla general pueden ser expresadas en un lenguaje comprensible para todos.

Albert Einstein

En los pocos centímetros cuadrados de un rostro, un hombre encuentra sitio para los rasgos de todos sus antepasados, para las expresiones de toda su vida y para sus deseos.

Ralph W. Emerson

4.1 Seguimiento

Llegados a este punto; podemos decir que hay una gran variedad de técnicas que puedo aprovechar para la implementación de este proyecto; sin embargo, nos decantaremos por otro camino, que explicaremos en breves.

Estoy de acuerdo, como muchos autores, quienes piensan que estimar el movimiento trabajando con mecanismos estadísticos, es sumamente dificultoso; ya que estimar el futuro movimiento de un sujeto es impredecible y está lleno de incertidumbres. Hay que tener cuenta; además, que trabajamos con imágenes anteriores e imágenes actuales; por tanto, se podría decir que en realidad no se está prediciendo el futuro, sino el presente del movimiento.

En primer lugar, vamos a ver a la imagen como una matriz de números, en otras palabras, como realmente es. Teniendo en cuenta esto, la búsqueda de un objeto (llame se rostro) dentro de una imagen, no es más que una búsqueda matricial.

Pero en la práctica hacer esto es mucho más complicado, dado que si buscamos el rostro de una forma exhaustiva en toda la imagen, el esfuerzo computacional es muy elevado.

Lo primero que tendremos que hacer es preparar la imagen con el objetivo de reducir el coste computacional en las siguientes etapas, una forma de hacerlo es pasar de una imagen a color a una en blanco y negro para trabajar con una escala de grises y así facilitar la comparación; ya que, lo que se quiere es rapidez en la detección de rostros, como se muestra en la imagen.



Figura 40: Seguimiento de rostros

4.1.1) Primera Solución

Se recomienda ver la figura 41 que esta al final de ésta sección, para facilitar su comprensión.

El objetivo de plantear una primera solución, no es más que explicar el funcionamiento del seguimiento de una forma sencilla, luego explicaremos (en una segunda solución) los cambios realizados para conseguir un óptimo funcionamiento. En realidad lo ideal sería que el proyecto funcionara con ésta primera solución; ya que da el mejor resultado en términos de eficiencia, el problema es que el coste computacional es tan elevado, que hace necesario cambiar algunos aspectos con el fin de hacerlo mucho más rápido sin perder su eficiencia o al menos no drásticamente.

Partiremos desde el punto en que cogemos una imagen en donde se ha detectado un rostro, la analizamos y guardamos la posición en el cual se encuentra el rostro. En la siguiente iteración tendremos una "imagen actual" en la cual aplicaremos la estimación de movimiento.

Primer paso:

Lo primero que hacemos es delimitar una zona de búsqueda dentro de la "imagen actual". En la figura, el cuadro rojo representa al rostro detectado en la primera iteración y contiene la posición inicial del mismo (es decir, la posición del rostro obtenido en la "imagen anterior" y representado en la "imagen actual"), lo que tenemos que hacer ahora es buscar hacia donde se ha movido el rostro en la "imagen actual", y para ello hemos delimitado una zona de búsqueda.

En un primer momento, se podría pensar que para implementar ésta solución necesitaríamos de cuatro bloques, los cuales buscarán si el movimiento ha sido hacia la izquierda, derecha, arriba, abajo e incluso una combinación de ellos.

Para resolver éste inconveniente, en nuestro código existe la variable desp.

desp: (desplazamiento) delimita la zona de búsqueda

Lo que en realidad hacemos es coger las posiciones (x, y) del rostro (bloque rojo) y luego desplazarla hacia la izquierda y hacia arriba un valor "desp" obteniendo (x-desp, y-desp) como la posición de inicio en la cual comenzaremos la búsqueda de la nueva posición del rostro. Las dimensiones de la zona de búsqueda son de $(2 * desp + 1)$ tanto en el eje X como en el eje Y.

Segundo Paso:

Una vez delimitado la zona de búsqueda se procederá a buscar la nueva posición del rostro, teniendo como punto de partida (x-desp, y-desp). Luego de ubicarnos en ésta posición seleccionaremos un posible candidato a rostro (éste paso se representa en la figura como un rectángulo de color gris), cuyo tamaño será igual al del rostro (rectángulo rojo).

En el instante de efectuar la comparación, entre el rostro y el supuesto rostro (rectángulos rojo y gris respectivamente) tenemos que hacer un recorrido a ambos, con ésta finalidad cogeremos un pixel de cada imagen y la restaremos, luego hallaremos su valor absoluto, el resultado de ésta operación será almacenada en una variable, el cuál irá acumulando el resultado de la diferencia de pixeles durante todo el proceso de comparación.

Al final nos quedaremos con la posición del "rostro" que proporcione la menor diferencia de pixeles, en otras palabras, con aquella que tiene más probabilidades de ser el rostro que estamos buscando.

Tercer Paso:

Iremos recorriendo la zona de búsqueda realizando lo explicado en el segundo paso e incrementando un pixel en cada iteración hasta que converja, o en otras palabras, se alcance la posición final (x+desp, y+desp).

Comentarios Importantes:

Hay que destacar que éste método sería el más apropiado para buscar el rostro en la "imagen actual"; ya que, hace una búsqueda exhaustiva píxel a píxel a lo largo de toda la zona de búsqueda. Sin embargo, tenemos que tener en cuenta la influencia del tamaño de la zona de búsqueda en el programa, a continuación citaremos, las principales consecuencias al variar el tamaño de la zona de búsqueda:

Zona de búsqueda grande

- Tiene mayor alcance
- Se minimiza el error
- Es más resistente a los cambios bruscos tanto de dirección como de velocidad.
- El esfuerzo computacional es elevado, por tanto es muy lento.

Zona de búsqueda pequeña

- Tiene menos alcance.
- Si el rostro se sale de la zona de alcance, por ejemplo si se mueve más rápido, entonces el error aumenta.
- Es más rápido; aunque no lo suficiente, como para considerarlo como ventaja.

Representación Gráfica

IMAGEN

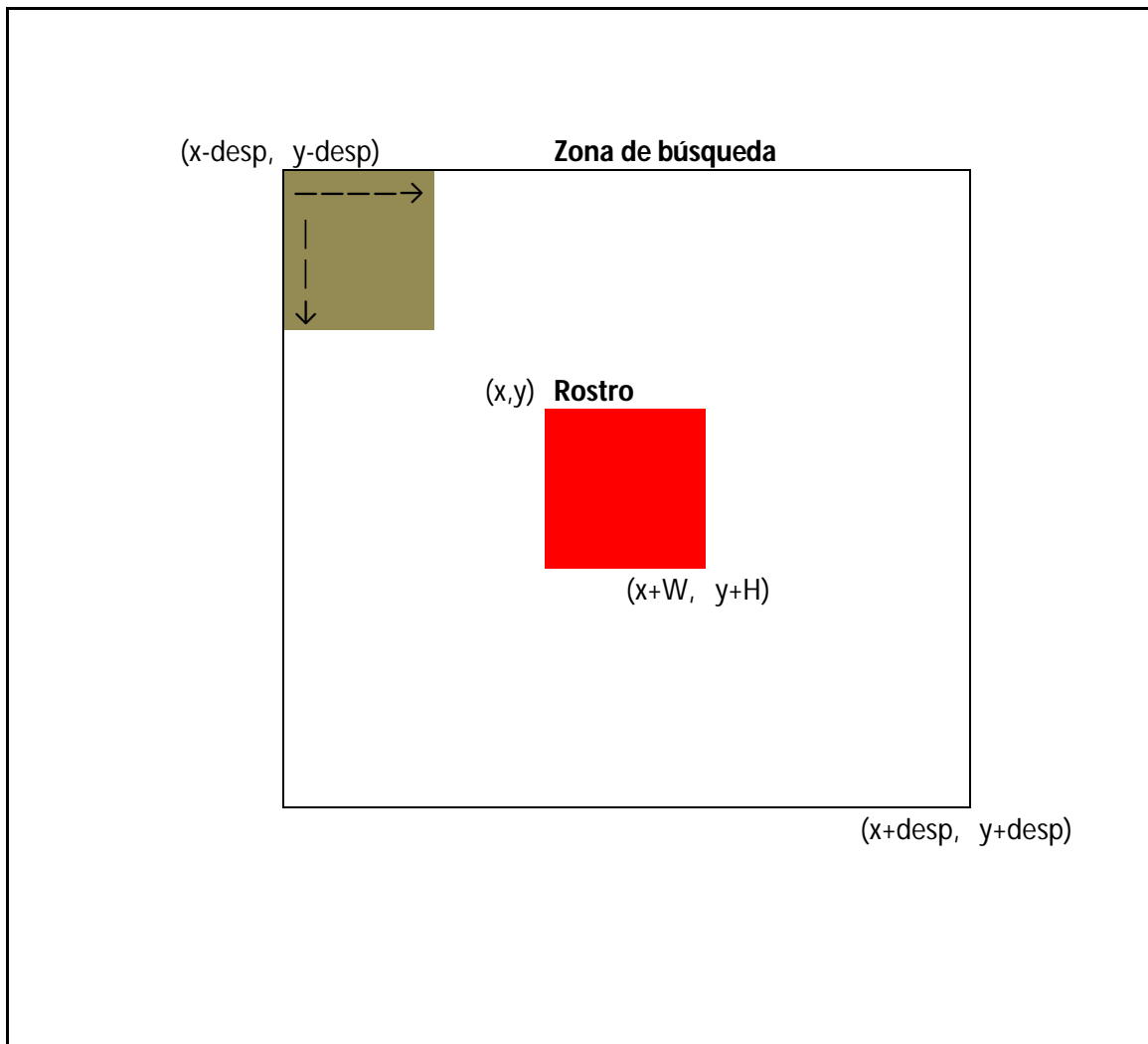


Figura 41: Seguimiento del rostro

Resumen de la Primera Solución:

- Como vemos en la imagen, el cuadrado gris se va desplazando pixel a pixel hacia la derecha y hacia abajo, de esta forma recorreremos la zona de búsqueda.
- Cada que desplazamos, cogemos un nuevo cuadrado gris, el cual será utilizado para compararlo con el cuadrado rojo (el rostro).
- Al comparar el cuadro gris con el rojo, lo que en realidad hacemos es recorrer los rostros (ambos cuadros) y coger un pixel de cada uno de ellos, para luego restarlos.
- Durante el proceso guardamos el resultado de la diferencia entre todos los píxeles en una variable que la va acumulando en un único resultado.
- Al finalizar nos quedamos con el cuadro gris que tenga el menor resultado; o sea, aquella que es la más parecida al cuadro rojo.

4.1.2) Segunda Solución

Tomamos como base la solución anterior; sin embargo, añadiremos a nuestro código dos variables con la finalidad de incrementar la velocidad y que el coste computacional sea menor.

1) Variable saltos:

Saltos: número de píxeles que saltamos dentro de la zona de búsqueda, con la finalidad de acelerar el recorrido; aunque, si los saltos son muy grandes, tendremos la desventaja de que aparezcan errores con mayor probabilidad.

Se podría decir que ésta solución es una optimización de la solución anterior ya que si los saltos = 1, entonces estaríamos haciendo la primera opción.

Delimitamos una zona de búsqueda dentro de la imagen como lo hacíamos anteriormente; pero al momento de recorrer la zona lo hacemos dando saltos que podrán ser grandes o pequeños.

La variable "saltos" y el tamaño de la zona de búsqueda (que depende de la variable "desp"; véase figura 42) están fuertemente relacionados.

Para un buen funcionamiento se recomienda que:

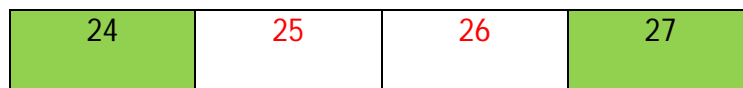
$$\text{desp} = 2 * \text{saltos}$$

Vamos a poner algunos ejemplos para que ésta situación sea más apreciable.

Ejemplo 1:

Supongamos que elegimos inicializar las variables de la siguiente manera:

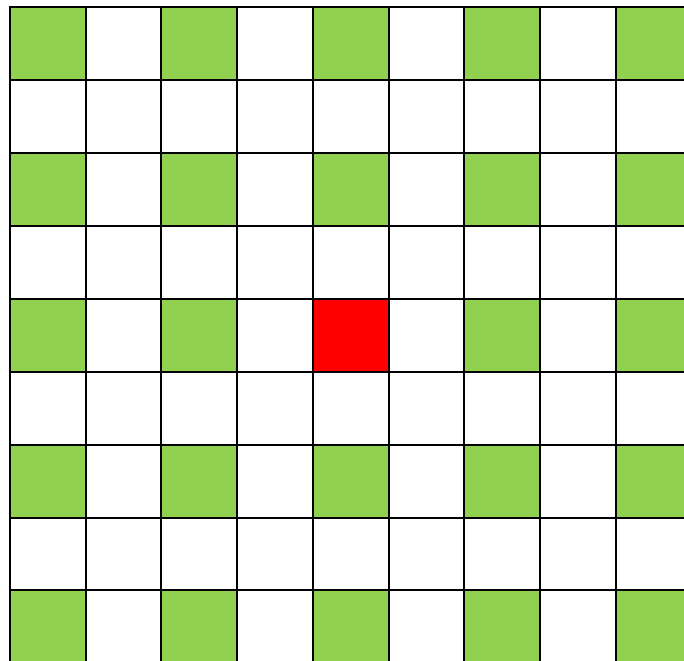
- saltos = 2,
 - desp = 4,
- La variable desp = 4; quiere decir que la zona de búsqueda se extiende cuatro píxeles en todas las direcciones, proporcionando un tamaño de 9 X 9, para éste ejemplo.
- La variable saltos = 2; por tanto los saltos tendrán un tamaño de un píxel por salto. Si por el contrario, "saltos" tiene un valor de 3, esto significa que aumenta obviamente de 3 en 3; pero aunque, podría parecer erróneo sólo salta 2 píxeles. Veremos un ejemplo, suponiendo que estamos en el píxel 24 y si saltos = 3, el próximo píxel sería el 27, pero como se aprecia en la figura sólo saltamos 2 píxeles (el 25 y el 26).



- En la siguiente figura se representan un cuadro rojo que es la posición del primer píxel del rostro, más veinticuatro cuadrados verdes y que en conjunto son las veinticinco posiciones en donde se buscará el posible movimiento del rostro.
- Como podemos apreciar la búsqueda, es geométricamente perfecta ya que toma en cuenta los posibles movimientos y presenta dos zonas, la primera que rodea a la posición original por si el movimiento se realiza de una forma normal y la segunda zona mucho más alejada, por si el sujeto se mueve más rápidamente; además también toma en cuenta si el sujeto no se mueve o realiza movimientos inapreciables quedándose en el mismo sitio.

Una vez aclarado estos puntos, veremos su funcionamiento en conjunto:

Zona de búsqueda



9X9

- Sin importar el valor de "saltos", siempre y cuando se cumpla que "desp" adopta el valor de "saltos" multiplicado por dos; entonces estas dos zonas estarán presentes, lo único que se diferenciarán es que tendrán una mayor separación, marcado por el valor de "saltos" obviamente.
- En conclusión éste método tiene un coste de veinticinco comparaciones por píxel, da buenos resultados, pero en ocasiones no es tan rápido como quisiéramos.

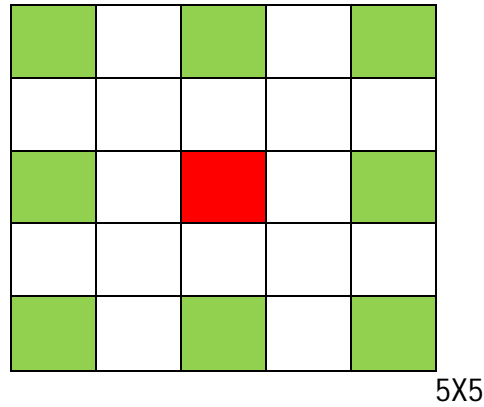
Ejemplo 2:

Si queremos que el programa sea más rápido entonces se recomienda que:

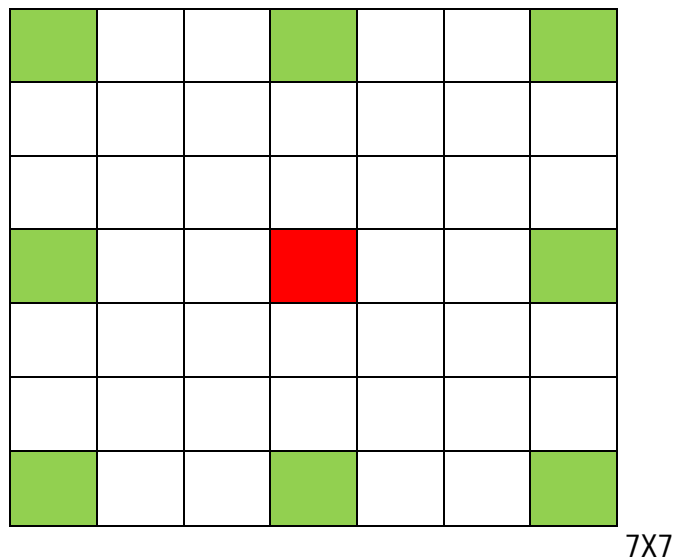
$$\text{desp} = \text{saltos}$$

- Independientemente del valor que adopten, éste método tendrá un coste de nueve comparaciones por píxel, veamos por ejemplo:

a) Si saltos = desp = 2, entonces:



b) Si saltos = desp = 3, entonces:



- A diferencia con el primer ejemplo, ésta vez sólo tenemos una única zona que rodea al píxel rojo, por tanto nuestro alcance y capacidad de percepción disminuye. Y a diferencia del apartado (a), podemos apreciar que la zona de búsqueda se encuentra más apartada del rectángulo rojo, eso quiere decir que tenemos más alcance, pero que no seremos capaces de detectar movimientos pequeños en la imagen.

- Hay que tener en cuenta la siguiente relación:

Saltos grandes

- o El seguimiento es más rápido.
- o Es más propenso a errores.
- o Tenemos mayor alcance al y mayor capacidad de seguimiento en casos en que el movimiento del rostro se efectúa con mayor velocidad

Saltos pequeños

- o Es más lento.
- o Minimiza los errores.
- o Es más sensible a pequeños movimientos de la cabeza.

Si hacemos una buena combinación entre los saltos y la variable "desp", como en el "ejemplo 1" los resultados serán mucho mejores, ya que podremos obtener un alcance muy alto y a la vez tener sensibilidad a los pequeños movimientos del rostro; sin embargo, estas ventajas lo pagamos con una mayor ralentización del programa.

2) Variable muestras_p:

muestras_p: Esta variable nos permite trabajar con un número medio de muestras deseado.

- Supongamos que tenemos una matriz de 1000 X 1000, si "muestras_p=10"; entonces trabajaremos con una media de 100 muestras (10X10).
- Lo que hacemos en realidad es dividir el tamaño original entre el número de muestras que queremos; sin embargo, en ocasiones el tamaño y el número de muestras que queremos es incompatible. Por ejemplo:
- Si tenemos un tamaño de 7 y muestras_p = 3 (como muestra la siguiente figura); entonces, los saltos entre píxel será de $7/3$ que es aproximadamente igual a 2.



4 píxeles de muestra de un tamaño de 7

- Esto quiere decir que tendremos cuatro muestras por fila y no tres como queríamos, es por eso que lo definimos como el número medio de muestras deseado. Esto como es de esperar no representa un problema crítico.
- Si ésta técnica la trasladamos al proyecto; en realidad, la utilizamos en el momento de hacer las comparaciones píxel a píxel entre el rostro y el supuesto rostro. Al trabajar con una muestra debidamente seleccionada (ya que coge puntos de diferentes partes de la imagen) de ambas, trae como consecuencia el aumento de la velocidad de procesamiento ya que hay menos comparaciones y esto produce una reducción del coste computacional.

Representación Gráfica

IMAGEN

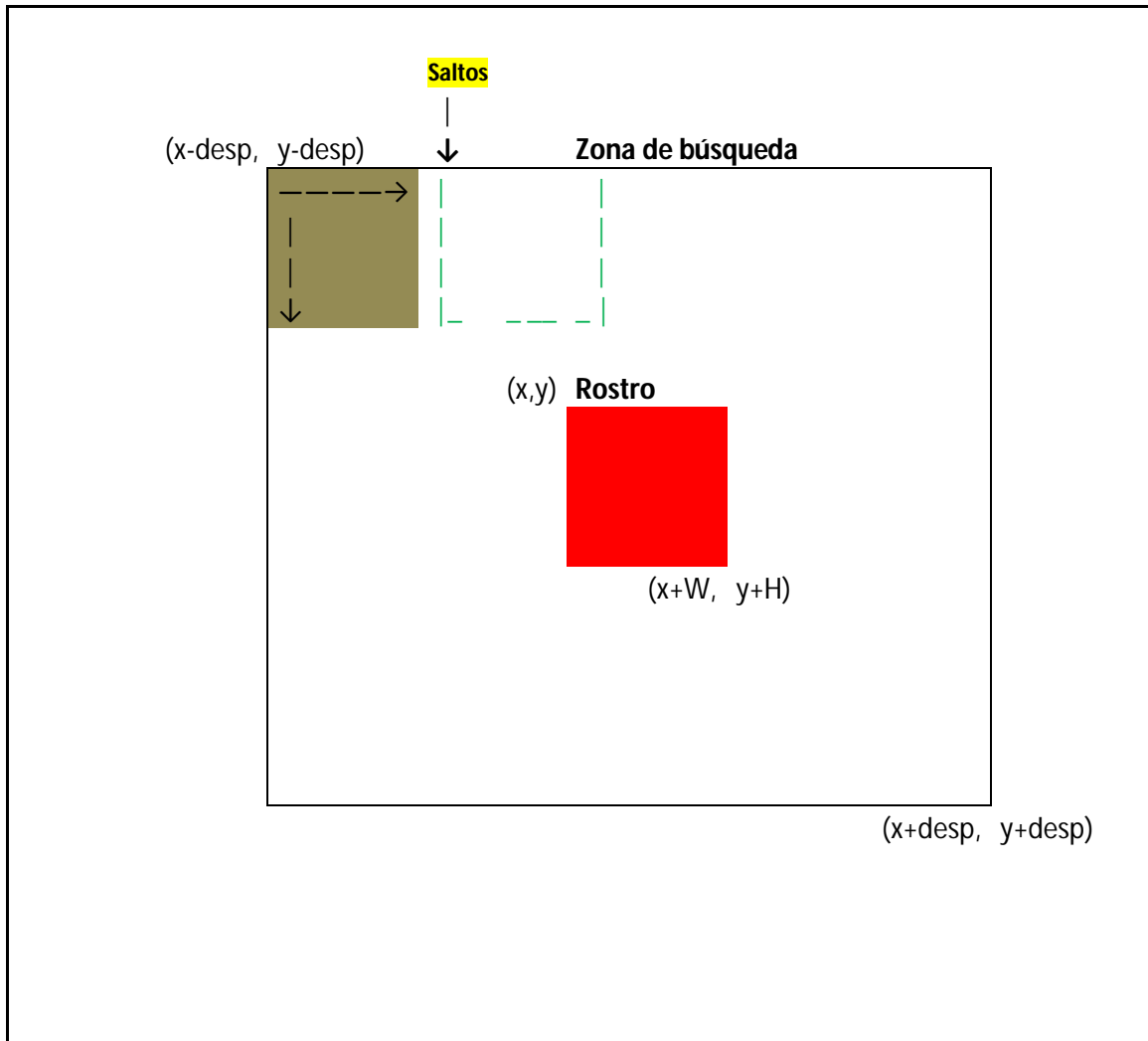


Figura 42: Seguimiento del rostro

4.1.3) Mejoras con esta propuesta

Mejoras en la velocidad:

En un video una persona puede variar su movimiento y la dirección en la que se mueve a una determinada velocidad, esto dificulta y hace más complicado la detección y su respectivo seguimiento.

Este problema lo resolvemos aumentando la zona de búsqueda para tener el mayor alcance posible; en otras palabras el rostro se ha movido más rápido y a recorrido mayor distancia dentro de la imagen, pero al tener un mayor alcance seremos capaces de detectarlo.

En un entorno en el cual, el sujeto se mueva rápidamente, sería bueno utilizar la técnica de las veinticinco comparaciones por píxeles, de esta manera nuestro alcance será mayor.

Esto origina un segundo problema, que es la ralentización del programa, dado que conlleva demasiado esfuerzo computacional. Para solucionarlo podríamos coger un número menor de píxeles como muestra, de esta forma ahorraremos más tiempo en el momento de comparar los rostros(ya que haremos menos comparaciones).

Como vemos existe una relación muy fuerte entre lo preciso que se quiere hacer el seguimiento y la velocidad con la que es capaz de desarrollarlo. Esto va a depender obviamente del campo en la que estemos orientándolo, que según sea se requerirá una u otra cosa.

Mejora en el zoom del rostro:

En un video, una persona puede alejarse o acercarse a la cámara, esto provoca un cambio en el tamaño del rostro que da la apariencia de empequeñecerse o agrandarse respectivamente. Eso quiere decir, que al momento de dibujar el rectángulo en el rostro, tiene que tener en cuenta esta variación y ser capaz de variar conjuntamente con el rostro.

Para solucionar éste inconveniente, utilizamos el refresco ya que el método cvHaarDetectObjects nos brinda una posición y el tamaño del objeto (rostro) que suponemos óptimos. Con éste método no sólo solucionamos el problema del zoom, sino también eliminamos los errores que hemos ido acumulando en el tiempo debido a saltos demasiado grandes o por que el rostro a salido del alcance de la zona de búsqueda.

Gestos y movimientos extraños del rostro:

En un video, la cantidad de expresiones faciales que hace una persona es elevada; así como también los giros de cabeza, sin ir más lejos el método cvHaarDetectObjects tiene muchos problemas a la hora de detectar a una persona que se encuentra de perfil; sin embargo, el método "reconoce" de la clase "Seguimiento" no tiene ningún problema dado que éste método esta basado en ver a la imagen como una matrices de números y las expresiones faciales no alterán de una forma grave el valor numérico del pixel que le corresponde a ese gesto en particular.

La estimación de movimiento se realiza de una forma óptima incluso en el caso de que el rostro gire 180 grados o más.

Saltos adaptativos

En un principio utilizamos esta variable, para controlar el número de comparaciones de píxeles entre el rostro y el supuesto rostro. Luego optamos por utilizar la variable "muestras_p" dado que independientemente del tamaño de la imagen, uno obtiene el número medio de muestras deseado.

Como dijimos, al referirnos al efecto zoom, una persona se puede acercar o alejar de la cámara en el video, esto con lleva otro problema. A parte, claro del cambio del tamaño del rectángulo, tenemos que tener en cuenta que la zona de comparación entre píxeles es mayor, eso quiere decir que haremos un mayor número de comparaciones.

Como solución a este problema introduciremos unos saltos entre pixeles en los rostros a comparar. A estos saltos le llamaremos saltos adaptativos ya que se irán adaptando al tamaño del rostro.

Si el rostro aumenta, entonces los saltos también lo harán, evitando un retardo innecesario y si disminuye, los saltos también disminuirán evitando que la comparación entre rostros sea poco eficiente.

Hemos tomado en cuenta que estos saltos no podrán tomar un valor menor o igual a cero; es más, sería conveniente que si esto pasara, el menor valor que adopte sería igual a dos para que asegure por lo menos un salto de pixel y evite errores al recorrer el bucle de comparación.

Añadiendo los saltos, hemos obtenido una gran velocidad a comparación con la primera solución; tanto, que si la imagen tiene un tamaño relativamente pequeño (video con resolución de 320 X 240) entonces el seguimiento es a tiempo real.

Problemas:

- El principal problema es que no teníamos un buen control de ésta variable, dado que había que tener en cuenta la posibilidad de que aumente de valor sin control y en lugar de facilitar la ejecución, la empeoraría.
- Por este problema principalmente se creyó conveniente utilizar la variable "muestras_p" dado que podíamos controlar independientemente del tamaño de la imagen, el número medio de muestras para comparar.

Resumen de la segunda solución:

- La segunda solución no es más que la optimización de la primera solución, ya que si la variable saltos es igual a uno (saltos=1) serán equivalentes.
- En ésta ocasión el cuadro gris ya no irá recorriendo la zona de búsqueda pixel a pixel, sino salto a salto.
- Esto significa que tendremos menos cuadros grises que comparar con el cuadro rojo (rostro buscado), por tanto será más rápido.
- Añadimos la variable muestras_p, ésta variable nos permite trabajar con un número medio de muestras deseado, con la finalidad de aumentar la velocidad de la iteración en el momento de la comparación entre pixeles.

4.2 Descripción general del proyecto

En este apartado , veremos un resumen general del proyecto, desde el momento en que la cámara coge una imagen hasta que el video finaliza, obviamente mi intención no es repetir todo lo dicho anteriormente, más bien dar un punto de vista diferente.

Lo que a continuación se presentará será la ejecución del programa paso a paso, pero sólo de los puntos más importantes, no volveré a explicar como se hace la estimación de movimiento, sólo lo mencionare.

- Paso 1: partimos de un video previamente grabado, al que analizaremos.
- Paso 2: capturamos una imagen en cada instante de tiempo y pasamos de una imagen a color a una imagen en blanco y negro con la finalidad de hacer más rápido el programa.
- Paso 3: activamos la función `cvHaarDetectObjects` para que detecte el rostro dentro del frame. Si no lo detecta, entonces coge el siguiente frame y así hasta que lo detecte.
- Paso 4: una vez que se ha detectado el rostro, procedemos a averiguar la posición y el tamaño de éste, luego lo guardamos para la siguiente iteración.
- Paso 5: dibujamos un rectángulo de color rojo en el rostro, hasta aquí acabamos con ésta iteración y pasamos a la siguiente en donde se efectuará la estimación de movimiento (el seguimiento).
- Paso 6: en este momento ponemos en marcha la función `reconoce` de la clase `Seguimiento`, el cual nos da la nueva posición del rostro, aplicando una estimación de movimiento, técnica que ha sido explicada durante todo el capítulo 4.
- Paso 7: dibujamos un rectángulo de color amarillo en el rostro, señal de que el seguimiento va correctamente.
- Paso 8: continuamos, con el seguimiento (pasos 6 y 7) hasta que el temporizador nos indica que es tiempo de hacer un refresco, esto lo hacemos con la finalidad de que los errores de estimación no se propaguen en el tiempo.
- Paso 9: al estar activado el refresco, el programa llama a la función `cvHaarDetectObjects`; si detecta el rostro, entonces refrescamos (obtendremos un vector de posición "sin errores de estimación"); en caso contrario, continuaremos haciendo el paso 8.
- Paso 10: se repite el paso 8 y 9 hasta que el video haya finalizado.

CAPITULO V

Resultados Experimentales



No he fracasado. He encontrado 10,000 soluciones que no funcionan.
Benjamin Franklin

5.1 Primer Experimento: video de 320 X 240

Los experimentos que realizamos a continuación, están hechos en un entorno cerrado, es decir dentro de una habitación, aunque cada experimento trata de comprobar el buen funcionamiento del programa a las distintas pruebas que se le realizaran.

El primer experimento será el más sencillo de todos , para esto contaremos con un video de 320 X 240, en el cual se verá a una persona haciendo diferentes movimientos con la cabeza, con el propósito de ver el comportamiento del programa con éste tamaño de video.

Antes de empezar con el experimento, haremos mención a ciertas consideraciones que hay que tener en cuenta, además haremos un recordatorio de los parámetros principales, que influyen en el programa, cada uno de éstos parámetros variará de acuerdo a la situación a la que ésta es sometida, por esta razón es preciso que el usuario comprenda su importancia y su funcionamiento.

Cada vez que utilicemos la función cvHaarDetectObjects dibujaremos un rectángulo de color rojo encerrando el rostro, eso significa que es la primera vez que se encuentra un rostro o que se ha producido un refresco del mismo; en cualquier caso, la posición del rostro que obtendremos es "la ideal";sin embargo, como se verá en el transcurso de los experimentos, esto no siempre es así.

Cuando utilicemos el metodo reconoce de la clase Seguimiento, el rectángulo será de color amarillo. La posición del rostro que obtengamos, dependerá mucho al buen funcionamiento del método cvHaarDetectObjects; ya que el método seguimiento seguirá aquella imagen que se encuentre dentro del rectángulo rojo, eso significa que si en lugar del rostro se encuentra una oreja, entonces se hará el seguimiento a la oreja. Además, hay que recalcar que el seguimiento se basa en la búsqueda de la mejor imagen contando unicamente con una muestra de la imagen original,por tanto nunca llegará a ser cien por ciento identicas

Los parámetros que influyen en el programa son los siguientes:

desp: por defecto, tendrá como valor el doble del parámetro saltos, esto lo hacemos con la finalidad de coger cinco muestras en cada eje, de esta manera tendremos un total de veinticinco muestras. Si se desea variar este parámetro, se recomienda mirar más a fondo su funcionamiento, en el capítulo 4.

saltos: su valor es de cuatro, esto me dice que en realidad entre el primer pixel y el siguiente pixel, habrán tres pixeles libres que equivalen al salto.

muestras_p: en este caso su valor es de doce, y eso significa que cogerá doce píxeles en el eje X y doce del eje Y, por tanto obtendrá una muestra de 144 píxeles por muestra, tanto del rostro como del supuesto rostro a comparar.

Tiempo: es el tiempo de refresco; en este caso su valor es de 50 y quiere decir que llamamos a la función cvHaarDetectObjects cada 50 frames.

Primer Experimento:

En la siguiente tabla podemos ver las características del video:

	Características del video
Dimensiones	320 X 240
Duración	12 segundos
Formato	avi

Tabla 1: características video 1

Las variables que determinan el mejor comportamiento del programa están inicializadas de la siguiente manera:

Variables	Valor Numérico
Salto	4
desp	8
muestra_p	12
tiempo	50

Tabla 2: variables del video 1

Resultados:

La detección del rostro en el video es inmediata, se aprecia que los rectángulos cambian de color y van alternando del rojo al amarillo y viceversa, además el seguimiento es rápido y muy bueno.

En la figura 43 se puede apreciar una muestra de cómo el seguimiento es capaz de detectar el rostro.



Figura 43: Seguimiento I

Por las razones expuestas anteriormente, las conclusiones que podemos llegar con este video son las siguientes:

- Duración aproximada del seguimiento es de 12 segundos.
- Prácticamente a tiempo real ya que es comparable a la duración del video original.
- El seguimiento es muy bueno; ya que se consigue rapidez y eficiencia tanto en los giros de cabeza, como en los acercamientos y alejamientos del mismo.

5.2 Segundo Experimento: video de 640 X 480

El video presenta las siguientes características:

	Características del video
Dimensiones	640 X 480
Duración	1:05 minutos
Formato	avi

Tabla 3: características video 2

A comparación con el primer video, éste tiene un formato mucho mayor, además el tiempo del video también es mayor. Dado que éste video tiene más tiempo, las pruebas que el programa tiene que pasar aumentan en número y dificultad ya que tiene que hacer un mayor número de comparaciones, a esto se añade que el número de muestras también crece.

Los datos que determinan el mejor comportamiento del programa están inicializados de la siguiente manera:

Variables	Valor Numérico
Salto	4
desp	8
muestra_p	8
tiempo	50

Tabla 4: variables video 2

Resultados:

El seguimiento tenía un reto mucho mayor con éste video, por las razones anteriormente explicadas, se apreció un retardo en el video provocado obviamente por el tamaño y la duración del mismo, sin embargo el objetivo de seguimiento del rostro es eficiente.

En éste video el individuo hace una serie de movimiento, que no se tomaron en cuenta en el primer video, por tanto la dificultad es mayor, además de un pequeño incremento de velocidad en sus movimientos, por éstas y demás razones, las conclusiones que podemos llegar con este video son las siguientes:

- Duración aproximada del seguimiento es de 2 minutos y 6 segundos, esto duplica la duración del video original, por tanto si quisiéramos que el programa sea más rápido sería preferible optar por una resolución de 320x240.
- El seguimiento no es perfecto, sobre todo por el retardo apreciable durante la ejecución del video, pero da buenos resultados en la mayoría de los casos.

- Al tener unas dimensiones mucho más grandes tarda más al momento de efectuar el seguimiento; ya que la zona de búsqueda es mucho más grande y por lo tanto comparamos más píxeles, además el tamaño del rostro es mayor, esto trae como consecuencia que la muestra sea también mayor (la muestra tiene más píxeles).
- Los distintos gestos, como el movimiento de la cabeza y demás no le afectan de manera irremediable, gracias al refresco los errores que íbamos arrastrando desaparecen.
- En éste video podemos apreciar los errores que comete la función cvHaarDetectObjects y por causa de este error el seguimiento será erróneo hasta el próximo refresco.
- Para evitar este error hay que tener cuidado a la hora de elegir el tiempo de refresco, no tendrá que ser muy grande ni muy pequeño.

En la figura 44 podemos apreciar el resultado de aplicar el seguimiento en el segundo video, además se ve el buen funcionamiento de éste sobre el video, al detectar perfectamente el rostro y enmarcarlo con un rectángulo amarillo.

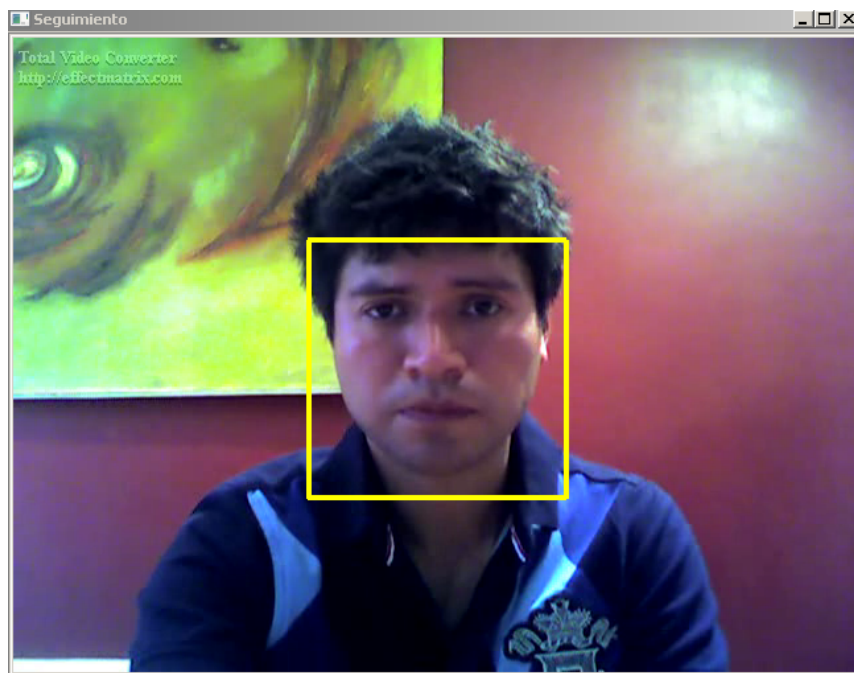


Figura 44: Seguimiento II

5.3 Tercer Experimento: cambios de iluminación

El video presenta las siguientes características:

	Características del video
Dimensiones	640 X 480
Duración	22 segundos
Formato	Avi

Tabla 5: características video 3

Este video es mucho más parecido al segundo video que al primero; ya que son de similares características; sin embargo, éste video tiene como objetivo comprobar el funcionamiento del seguimiento en los distintos cambios de iluminación, para esto, durante la realización del video vamos encendiendo y apagando una lámpara, con el propósito de ir modificando la iluminación y de ésta manera alterar el video en sí. Eso quiere decir que las pruebas que el programa tiene que pasar serán mucho más difíciles que las anteriores; ya que existe la posibilidad de que en algún momento nos quedemos totalmente a oscuras y no tengamos ninguna referencia para la detección y seguimiento del rostro.

Los datos que determinan el mejor comportamiento del programa están inicializados de la siguiente manera:

Variables	Valor Numérico
Salto	8
desp	16
muestra_p	8
tiempo	50

Tabla 6: variables del video 3

Resultados:

El seguimiento tenía un reto mucho mayor con éste video, por las razones anteriormente explicadas, se apreció un retardo en el video provocado obviamente por el tamaño y la duración del mismo, sin embargo el objetivo de seguimiento del rostro sigue siendo eficiente.

En éste video el individuo hace una serie de movimiento con la dificultad añadida de ir variando la iluminación de la habitación. Los resultados de este experimento son muy buenos , incluso mejor de lo que se esperaba y como muestra de ello se representa la figura 45, en la cual el seguimiento del rostro es eficiente en un entorno casi sin iluminación.

En la figura 45 podemos apreciar el resultado de aplicar el seguimiento en el tercer video, además se ve el buen funcionamiento de éste sobre el video, al detectar perfectamente el rostro y enmarcarlo con un rectángulo amarillo.

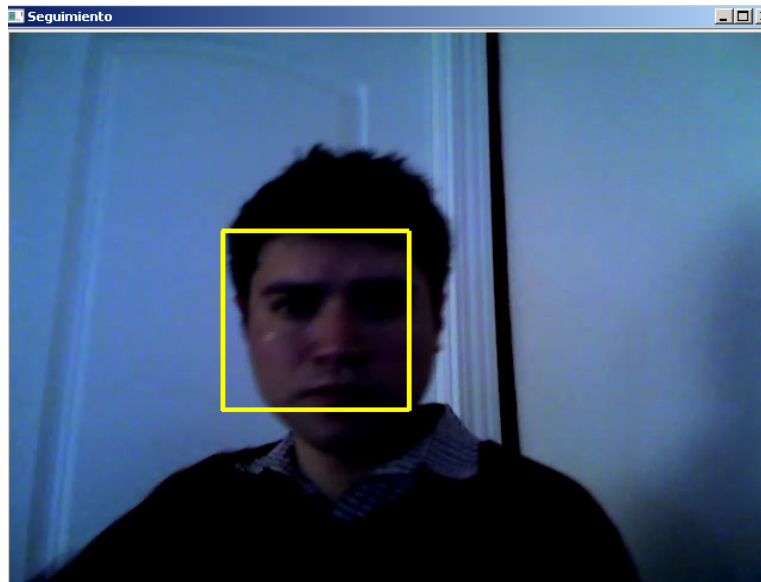


Figura 45: Seguimiento III

Por las razones expuestas anteriormente, las conclusiones que podemos llegar con este video son las siguientes:

- Duración aproximada del seguimiento es de 39 segundos.
- Al tener unas dimensiones mucho más grandes tarda más al momento de efectuar el seguimiento; ya que comparamos más píxeles.
- Los cambios de iluminación no parece afectarle.
- El seguimiento no es perfecto, pero da buenos resultados en la mayoría de los casos.

5.4 Cuarto Experimento: velocidad

El video tenía las siguientes características:

	Características del video
Dimensiones	640 X 480
Duración	16 segundos
Formato	Avi

Tabla 7: características video 4

Este video es mucho más parecido al segundo y al tercer video; ya que son de similares características; sin embargo, éste video tiene como objetivo comprobar el funcionamiento del seguimiento cuando un individuo se mueve a gran velocidad y cuando hace cambios bruscos de sentido. Eso quiere decir que las pruebas que el programa tiene que pasar serán mucho más difíciles que las anteriores; ya que si el individuo se mueve más rápido, tendremos que tener un mayor alcance de detección, por tanto la zona de búsqueda crece a comparación con los anteriores videos, además en este tercer video, el seguimiento también tendrá que afrontar con los problemas que caracterizaban al segundo video.

Los datos que determinan el mejor comportamiento del programa están inicializados de la siguiente manera:

Variables	Valor Numérico
Salto	20
desp	40
muestra_p	8
tiempo	30

Tabla 8: características video 4

Resultados:

El seguimiento tenía un reto mucho mayor con éste video, por las razones anteriormente explicadas, se apreció un retardo en el video provocado obviamente por el tamaño y la duración del mismo, sin embargo el objetivo de seguimiento del rostro es aceptable.

En éste video el individuo hace una serie de movimiento de forma rápida, además también realiza cambios bruscos de dirección, Los resultados de este experimento son aceptables, no se consigue el continuo seguimiento del rostro debido a la alta velocidad del movimiento, pero esto se compensa; en parte, con refrescos, que en el caso de ser óptimos, nos permitirán realizar un mejor trabajo.

En las figuras 46 y 47 podemos apreciar el resultado de aplicar el seguimiento en el cuarto video, además se ve el buen funcionamiento de éste sobre el video, al detectar perfectamente el rostro y enmarcarlo con un rectángulo amarillo.

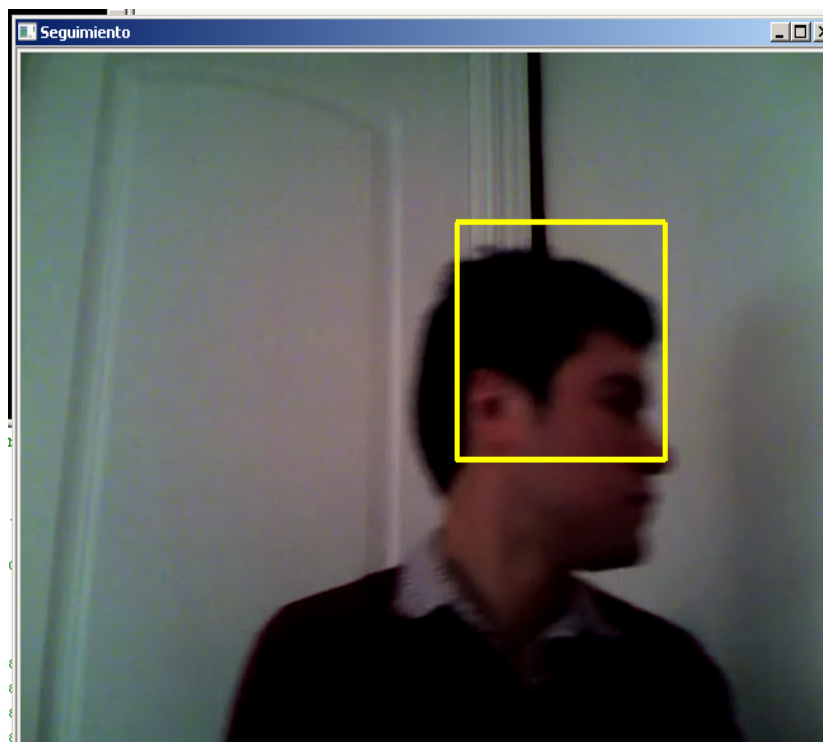


Figura 46: Seguimiento IV

En la figura 47, podemos apreciar la recuperación del seguimiento tras un breve periodo de tiempo, en el cual, debido a la velocidad del movimiento, perdió las referencias originales de la posición del rostro, lo que provocó errores al momento de la detección y seguimiento del mismo, sin embargo gracias a un buen refresco, fue capaz de continuar con el seguimiento recuperándose de los errores producidos anteriormente.

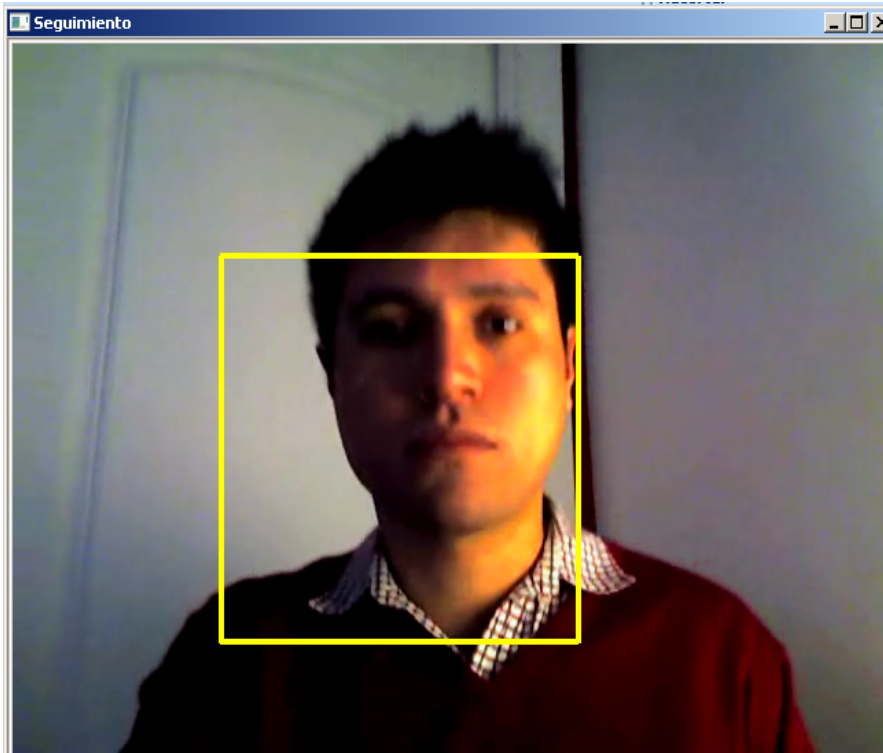


Figura 47: Seguimiento V

Por las razones expuestas anteriormente, las conclusiones que podemos llegar con este video son las siguientes:

- Duración aproximada del seguimiento es de 35 segundos.
- En este caso la velocidad es mucho mayor que los casos anteriores, el resultado es que el seguimiento funciona con buenos resultados, pero si el rostro se mueve con rapidez durante mucho tiempo, el seguimiento pierde la referencia con la posición original y esto provoca errores, hasta un futuro refresco.

CAPITULO VI

Conclusiones



La vida es el arte de sacar conclusiones suficientes a partir de datos insuficientes.
Samuel Butler (novelista)

6.1 Conclusiones

- Lo primero que me gustaría decir, es que el proyecto en su inicio fue pensado para que la imagen captada, fuera a tiempo real desde una cámara; sin embargo, esto no fue posible debido a los problemas de compatibilidad entre las OpenCv y el windows visual. La solución fue hacer un video y posteriormente trabajar con él.
- Una vez hecho el video, éste tiene que tener un formato “.avi”, en caso contrario no funciona, al parecer aquí influyen varios factores, desde el tipo de cámara que se esté utilizando, el sistema operativo donde se trabaje y su compatibilidad con las OpenCV.
- En lugar, de tener saltos adaptivos, cogemos una cantidad de muestras constantes, independientemente del tamaño del rostro. Es preciso aclarar que no es igual coger una muestra de 100 píxeles de un rostro cuyas dimensiones son por ejemplo de 20x20 que de otro que tiene unas dimensiones de 40x40; ya que, la muestra representará mejor al primeo de ellos (al de menor tamaño). Por ésta razón es preciso saber las dimensiones del video con el que estamos trabajando, para elegir un tamaño de muestra acorde con el video, esto evitará posibles errores al momento de efectuar la búsqueda y la comparación del mismo.

A continuación, les presentaremos las principales ventajas y limitaciones que hemos sido capaces de observar durante el desarrollo y depuración del proyecto.

6.1.1) Ventajas

- Cada vez hay procesadores más rápidos; si bien he cierto uno de nuestros mayores problemas es el gran coste computacional que tiene éste proyecto, al querer que la detección y el seguimiento del rostro sean lo más rápido posible, tenemos que ajustar el programa e incluso aumentar la probabilidad de error con el fin de ganar en velocidad. Esto lo hacemos obviamente porque estamos muy limitados por el procesador, ya que se le exige demasiado; sin embargo a medida que pasa el tiempo los procesadores son cada vez más potentes y éste problema dejará de ser al fin un inconveniente, ya que incluso ahora y gracias a esa evolución se ha logrado sacar adelante éste proyecto, tal vez impensable hace una década atrás.

- Si la imagen tiene un tamaño relativamente pequeño, entonces se puede considerar que el seguimiento se ejecuta a tiempo real.
- El cambio de fondo en la imagen, no afecta al seguimiento, esto lo pudimos comprobar con el experimento número tres, en el cual cambiamos de iluminación y pasábamos de un entorno iluminado a uno oscuro y viceversa, además, en el momento de encender la lámpara, se producía un efecto flash en el video, por la saturación de los píxeles.
- Los gestos y movimientos extraños no afectan el seguimiento; con lo que respecta a los gestos, éstos no representará un problema, esto es debido a que al momento de efectuar la búsqueda, lo hacemos basándonos en una muestra del rostro original y tratando de hallar aquella imagen que más se parezca a esta muestra; es decir, las que tengan similares valores de píxeles, y los gestos no implican un cambio radical en el valor de los píxeles de la imagen, por esta razón decimos que no es un problema que afecte de manera irreparable. Los movimientos del rostro como se ha podido apreciar en los experimentos, es bastante aceptable, aunque eso también dependerá de la velocidad en que se hagan.
- Soluciona hasta cierto punto el problema de los cambios bruscos y los cambios de velocidad con que se mueve el rostro.
- El cambio de tamaño del rostro no es un problema; en los anteriores experimentos el sujeto se aproximaba y alejaba de la cámara, esto no representaba ningún tipo de problema, para la detección y seguimiento del mismo.

6.1.2) Limitaciones

- Fallos del método cvHaarDetectObjects, si éste método falla al momento de detectar el rostro, entonces arrastraremos durante un tiempo una serie de errores, hasta que se produzca un nuevo refresco.
- Si los saltos son de pixel en pixel, la ejecución del programa se torna demasiado lento, el procesador actual con el que contamos, no es capaz de trabajar a éste nivel, lo único que lograremos es colgar el procesador.
- Si los saltos son muy grandes, da la impresión de que el cuadrado que está siguiendo el rostro, baila en la imagen, sin ningún tipo de seguimiento.

- Si la zona de búsqueda es muy pequeño, sólo seríamos capaces de detectar pequeñas variaciones en la imagen; por tanto, si el individuo se mueve rápido, sería impensable que pudiésemos seguirlo.
- Si el tamaño de la imagen del video es grande, entonces tardará más debido al incremento en el número de comparaciones entre imágenes y al incremento de los píxeles en la muestra de cada imagen.
- Si el rostro se acerca a la cámara, esto produce un retardo; ya que el rostro aumenta de tamaño y como consecuencia amplía aún más el número de píxeles en la muestra de cada imagen.

6.2 Trabajos Futuros.

- Sería una opción interesante, aplicar este proyecto, en uno mucho más grande, el cual tenga como finalidad el seguimiento de varios rostros.
- Una posibilidad para que sea capaz de hacer el seguimiento a varios rostros, es implementándolo mediante procesos paralelos para evitar la ralentización.
- Actualizar el software es importante, pero cuando hablamos del uso de nuevas tecnologías, es mucho más importante actualizar el hardware, sobre todo si va orientado a la defensa nacional. En éste caso en concreto, tener un procesador rápido nos solucionaría el problema de ralentización del software, pudiendo ser más rápidos y eficientes.
- Como hemos visto antes, hay una gran variedad de técnicas para la detección y el seguimiento de rostros, un proyecto interesante sería obtener el tiempo que tarda cada una de ellas y si no hay mucha diferencia, entonces elegir la más sencilla; ya que como dije antes, el hardware puede solucionar esas diferencias.
- Éste proyecto pretende ser la base de futuras investigaciones, si se quiere orientar a la industria o a diversos campos obviamente sería recomendable modificarlo, para resaltar los puntos que más nos interesan. Por ejemplo si lo queremos orientar a la vigilancia, sería bueno tener en cuenta una variable de velocidad con la finalidad de que la cámara gire hacia el objetivo, también sería interesante saber la posición de la cámara más cercana, para enviarle la información y que siga con la función de vigilancia, así si un individuo está corriendo, es un posible sospechoso y estará vigilado por un trabajo en conjunto de cámaras "inteligentes".

CAPITULO VII

Presupuesto



Nada se construye en los plazos previstos ni sin salirse del presupuesto.

Ley de Keops

PRESUPUESTO INICIAL DEL PROYECTO

DESCRIPCIÓN	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
Director de proyecto: Ing. Edwin Vega Aquino	6 meses	2 000€/mes	12 000€
Licencias: Visual C++	6 meses		500€
Un ordenador			2 000€
Total			14 500€

CAPITULO VIII

Bibliografía



Por el grosor del polvo en los libros de una biblioteca pública puede medirse la cultura
de un pueblo.
John Ernst Steinbeck

Los libros son espejos: sólo se ve en ellos lo que uno ya lleva dentro
Carlos Ruiz Zafón

A continuación daré una pequeña reseña bibliográfica, sin ayuda de las cuáles, éste proyecto no habría sido posible, así que tengo que dar las gracias a todos aquellos científicos que viven por y para la ciencia, y muy en especial aquellos que mencionare en breve.

- [1] Computer vision with the OpenCV library; Gary Bradsky y Adrian Kaebler.
- [2] Manual original proporcionado por intel: *Intel® Open Source Computer Vision Library*.
- [3] Adaptive Probabilistic Visual Tracking with Incremental Subspace Update. David Ross, Jongwoo Lim, and Ming-Hsuan Yang.
- [4] EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. International Journal of Computer Vision 26(1), 63–84 (1998). MICHAEL J. BLACK, ALLAN D. JEPSON.
- [5] Online Learning of Probabilistic Appearance Manifolds for Video-based Recognition and Tracking; Kuang-Chih Lee y David Kriegman.
- [6] Eigenfaces for Recognition; Turk, Matthew A., Pentland, Alex P.
- [7] Face Recognition Using Eigenfaces, Turk, Matthew A., Pentland, Alex.
- [8] Learning OpenCV: Computer Vision with the OpenCV Library, Bradski, G., Kaebler, A.
- [9] Detección de Rostros en Imágenes Digitales usando Clasificadores en Cascada; Guevara, M., Echeverry, J., Ardila, W.
- [10] Decision-theoretic generalization of on-line learning and an application to boosting; Freund, Y., Schapire, R. A.
- [11] <http://ltilib.sourceforge.net/doc/html/compilers.html>, para información sobre LTI-LIB.
- [12] <http://vxl.sourceforge.net/>, para información sobre VXL.
- [13] <http://public.kitware.com/vxl/doc/release/books/core/book.html>, libro online de la VXL.
- [14] http://www.elpais.com/articulo/sociedad/robot/vision/artificial/recoger/fresas/elpepusoc/20100528elpepusoc_12/Tes, diario El País Madrid- España.

- [15] Human face detection in complex background. Patter Recognition, 1994; G. Yang and T:S:Huang.
- [16] Tracking in 3D: Image variability decomposition for recovering object pose and illumination. Pattern Analysis and Applications 1999. P:N: Belhumeur and G.D. Hager.
- [17] Recognizing facial expressions in image sequences using local parameterized models of image motion, International Journal of Computer Vision, 1997. M.J. Black and Y. Yacoob.
- [18] Análisis de expresiones faciales mediante visión por computador. Universidad politécnica de Madrid, 2004. J.M. Buenaposada Biencinto.
- [19] Video-based framework for face recognition in video. In Second Workshop on Face Processing in Video. 2nd Canadian conference on Computer and Robot Vision. D.O. Gorodnichy.
- [20] Face detection: A survey Computer Vision and Image Understanding, 2001. E. Hjelmas and B. Kee Low.
- [21] Fast object detection with oclussions. European Conf. on Computer Vision, 2004. Y. Lin, L. Liu and S. Fuh.
- [22] 2D Cascaded AdaBoost for eye localization, In the 18th International conference on Patter Recognition, 2006. Z. Niu, S. Shan, S. Yan, et al.
- [23] Recent advances in face detection, Cambridge 2004. H. Yang.
- [24] Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. Computer Vision and Image Understanding. Z. Zhu and Q. Ji.
- [25] Overview of the face recognition grand challenge. In Computer vision and Pattern Recognition, 2005. P.J. Phillips, P.J. Flynn, et al.
- [26] Procesamiento de Caras Humanas mediante Integrales Proyectivas, Tesis doctoral, Universidad de Murcia, 2007. Ginés García Mateos.