CHAPTER X

Sistema de atención visual para la interacción persona-robot

J. VEGA[1] y J.M. CAÑAS[1]

[1]Universidad Rey Juan Carlos, julio.vega@urjc.es, jmplaza@gsyc.es

Resumen. Los sistemas de visión son hoy en día uno de los elementos sensoriales más usados en robótica autónoma. Su principal dificultad radica en extraer información útil a partir de las imágenes capturadas y el campo visual pequeño de las cámaras normales. Los sistemas de atención visual y la visión activa ayudan a superarla. Por otro lado, cada vez más se demandan robots que interactúen con humanos. Basándonos en la interacción más básica entre humanos: *mirarse a la cara*, hemos construido un sistema cuyo campo de visión abarca más que el propio de la única cámara que empleamos, montada ésta sobre un cuello mecánico. El algoritmo mueve de manera inteligente el cuello mecánico con el objetivo de encontrar caras nuevas, seguir a las ya existentes y olvidar a aquéllas que ya no aparezcan en la escena. Para ello, se han seguido dos dinámicas concurrentes: la vida y la saliencia.
Palabras clave. Atención visual, seguimiento de caras, exploración de escena, interacción persona-robot.

Abstract. Vision systems are today one of the most often used sensory elements in autonomous robotics. Their main difficulty is to extract useful information from the captured images and the small visual field of normal cameras. Visual attention systems and active vision help to overcome it. On the other hand, are increasingly demanding robots that interact with humans. Based on the most basic interaction between humans: *looking at one another*, we have built a system whose field of vision encompasses more than the only camera we use, mounted on a pan-tilt unit. The algorithm intelligently moves this neck with the aim of finding new faces, follow existing and forget those that no longer appear on the scene. This algorithm

has been designed and implemented according to two competing dynamics: life and salience.
Keywords. Visual attention, face tracking, scanning the scene, human-robot interaction.

1 Introducción

La evolución de la robótica ha llevado en los últimos años a intentar introducir los robots en el ámbito más cercano del hombre. A dichos robots se les puede denominar *sociales*. Este acercamiento implica que los robots tengan determinadas características que el hombre considera necesarias para aceptarlo.

Uno de los principios de éstos es que sean capaces de percibir y entender las formas naturales de comunicación de los humanos. Y la primitiva más básica de interacción entre humanos es *mirarse a la cara*. Es por tanto interesante desarrollar técnicas que permitan al robot conocer la posición de las personas que están alrededor de él, y seguirlas en todo momento. De ahí que para una interacción *robot-humano* aceptable resulte indispensable un mecanismo de detección y seguimiento de caras *(Lang, 03)*. Esto permite una interacción más fluída con las personas, pues éstas perciben más natural al interlocutor robótico.

La cara proporciona información no sólo de la posición de la persona que está interactuando con el robot, también resulta indispensable para identificarla. Además, refleja su estado de ánimo, e incluso la intención del humano. Concretamente la parte de la cara más importante y que confiere más sentimientos son los ojos *(Yoshikawa, 06)*, la mirada de una persona.

Por otro lado, y de forma paralela, el uso de cámaras en los robots está creciendo continuamente. Éstas se han convertido en un sensor cuyo coste se ha ido abaratando en los últimos años y que potencialmente pueden proporcionar al robot una gran cantidad de información sobre su entorno. Sin embargo, la enorme cantidad de datos que vierten las cámaras no es fácil de procesar. Así, los mecanismos de atención del sistema de visión humana han sido fuente de inspiración para el diseño de los sistemas artificiales de atención visual, con el propósito de utilizar eficientemente los recursos computacionales.

En este trabajo se presenta un sistema de atención global para un robot móvil dotado de un cuello mecánico, al que va acoplado una sóla cámara. Este sistema realiza un procesamiento de las imágenes capturadas por la cámara para, mediante métodos basados en apariencia, detectar las posi-

bles caras humanas presentes en la escena, incluso más allá del campo visual instantáneo. Dado que una conversación es la máxima expresión de comunicación humana, nuestro sistema es capaz de *seguir con la mirada* a todas las personas que estén allí presentes, alternando la mirada entre todas ellas. Se establecen dos variables dinámicas, saliencia y vida, asociadas a cada cara. Su evolución temporal dirige el comportamiento del sistema. Así, éste continuamente da respuesta a dos preguntas: ¿Cuántas caras hay en la escena circundante al robot? y, ¿dónde están situadas?

Tras esta introducción, la segunda sección describe brevemente el estado del arte en lo que atención visual en robots se refiere. En la tercera sección describimos el diseño de nuestro mecanismo de atención: detección de caras, seguimiento, reparto de la mirada, exploración de la escena, y las dinámicas de saliencia y vida. La cuarta sección está dedicada a la implementación del sistema. Para probar el comportamiento y rendimiento del mismo se han llevado a cabo numerosos experimentos, cuyo resumen se refleja en la quinta sección. Finalmente, el artículo acaba con una serie de conclusiones y trabajos futuros.

2 Atención visual en robots

Dentro de la visión artificial un área de interés creciente es la *atención visual*. La atención es la fijación en uno o varios aspectos de la realidad y prescindir de los restantes; esto es, tener preferencia sobre ciertos objetos que sobresalen por alguna característica, ya sea por su color, forma, etc. La atención dispone de dos etapas claramente marcadas, la primera, considerada procesamiento previo, es aquélla en la que se extraen objetos -que cumplen determinadas características- dentro del campo visual; y la segunda, llamada atención enfocada, consiste en la identificación de esos objetos.

Dentro de la robótica autónoma es importante realizar un control de atención visual. Las cámaras de los robots proveen de un amplio flujo de datos del que hay que seleccionar lo que es interesante e ignorar lo que no; en esto consiste la atención visual selectiva. Existen dos vertientes de atención visual, la global (*overt attention*) y la local (*covert attention*). La atención local *(Tsotsos, 95)*, *(Itti, 01)*, *(Marocco, 02)* consiste en seleccionar dentro de una imagen aquellos datos que nos interesan. Y la atención global consiste en seleccionar del entorno que rodea al robot, más allá del campo visual actual, aquellos objetos que interesan, y dirigir la mirada hacia ellos *(Cañas, 05)*.

La representación visual de los objetos interesantes en los alrededores del robot puede mejorar la calidad del comportamiento del robot, así como la posibilidad de manejar más información a la hora de tomar sus decisiones. Esto plantea un problema cuando esos objetos no se encuentran en el campo de visión inmediato. Para solventar este inconveniente, en algunos trabajos se emplea visión omnidireccional; en otros, se utiliza una cámara normal y un mecanismo de atención global *(Itti, 01)*, *(Zaharescu, 05)*, que permite -de forma rápida- tomar muestras de un área de interés muy amplio. El uso del movimiento de la cámara para facilitar el reconocimiento de objetos fue propuesto por *(Ballard, 91)*, y se ha utilizado, por ejemplo, para distinguir entre diferentes formas en las imágenes *(Marocco, 02)*.

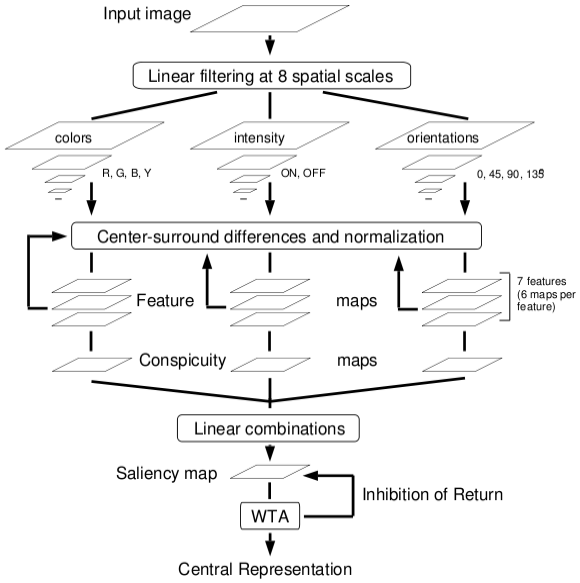


Figura 2.1: Esquema de formación del mapa de saliencia

Uno de los conceptos ampliamente aceptados en los trabajos del área es el de *mapa de saliencia.* Lo podemos encontrar en *(Itti, 01)*, como un mecanismo de atención visual local, independiente de la tarea particular a realizar, y formado por el conjunto de estímulos visuales que llaman la atención de la escena. En tal trabajo se considera púramente un modelo de "abajo a arriba" o *bottom-up*, donde -como podemos ver en la *figura 2.1*- en cada iteracción compiten los diferentes mapas descriptivos de la escena (según colores, intensidades u orientaciones) para a continuación fundirse en lo que denominan mapas de conspicuidad (uno por cada rasgo caracte-

rístico) y que finalmente conformarán un único y representativo mapa de saliencia.

3 Sistema de atención visual

El objetivo de nuestro sistema es realizar un seguimiento de atención sobre las caras de las personas presentes en la escena circundante al robot. Por tanto, se deben captar nuevas caras, repartir la mirada sobre las allí existentes y eliminarlas de la memoria una vez hayan desaparecido.

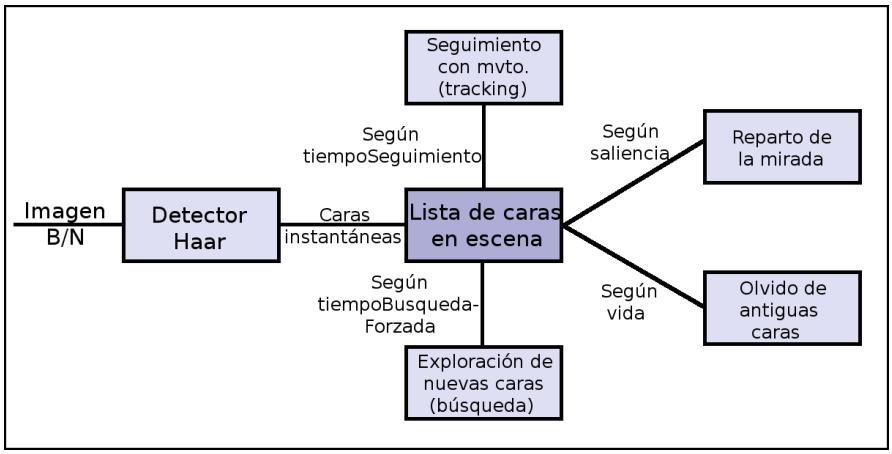


Figura 3.1: Diagrama de bloques del sistema

En esta sección veremos los distintos componentes en los que se basa el comportamiento del sistema desarrollado *(figura 3.1)*. Por un lado, el *detector de caras*, que es el encargado de identificar los rostros humanos que hay en la imagen en curso. La *memoria visual* permite ampliar el campo de visión a toda la escena circundante al robot, no sólo el campo visual instantáneo. En esta memoria se van situando todo el conjunto de caras detectadas, con ciertos atributos: posición en la escena, *saliencia* (para decidir dónde mirar en cada momento) y *vida* (para olvidar una cara que haya desaparecido de la escena). Además, tendremos un mecanismo de *seguimiento* de éstas con movimientos de la cámara, implementado como un controlador *P*; y otro mecanismo que nos permitirá *explorar* nuevas zonas desconocidas de la escena.

3.1 Detección de caras

El primero de nuestros componentes es el detector de caras. Se encarga de detectar las caras humanas existentes en la imagen actual que recibe de la cámara, vertiendo las posiciones de éstas dentro de la imagen.

Actualmente, las mejores técnicas en detección de caras son las basadas en apariencia. Éstas se fundamentan en la comprobación de las regiones de imagen con un modelo de cara para determinar si lo son o no. Este modelo debe ser aprendido previamente durante un proceso de entrenamiento, por lo cual también se suele conocer a estas técnicas como métodos basados en aprendizaje.

En general, los detectores basados en apariencia funcionan de modo similar, realizando una serie de pasos:

1.- *Búsqueda multiescala.* A partir de la imagen de entrada se obtiene una pirámide de imágenes con diferentes resoluciones.

2.- *Preprocesamiento y foco de atención.* Algunos métodos preprocesan las ventanas obtenidas.

3.- *Clasificación.* En este punto se determina por cada ventana si es una cara o no. En *(Yang, 02)* se explica este problema dentro de un marco probabilístico.

4.- *Aproximación de candidatos y postprocesamiento.* Tras la clasificación se obtiene un conjunto de posibles regiones válidas. En este paso los detectores agrupan regiones con un elevado solapamiento y eliminan candidatos poco fiables.

La calidad de la detección no sólo dependerá de la eficacia de cada una de las etapas mencionadas, puesto que siempre es necesario un modelo representativo y bien entrenado. Para el entrenamiento se emplean cientos de muestras a la misma escala, llamadas ejemplos positivos, y otras tantas imágenes aleatorias, ejemplos negativos.

Detector basado en filtros Haar en cascada AdaBoost

Éste es el detector de objetos que proporciona la biblioteca *OpenCV*[1] y es el que hemos utilizado. Este algoritmo fue propuesto inicialmente por Paul Viola y Michael Jones en *(Viola y Jones, 01)* y mejorado después por Rainer Lienhart y Jochen Maydt en *(Lienhart y Maydt, 02)*.

Trabaja internamente con imágenes en escala de grises (componente primero de nuestro sistema, *figura 3.1*). Requiere que primero se entrene

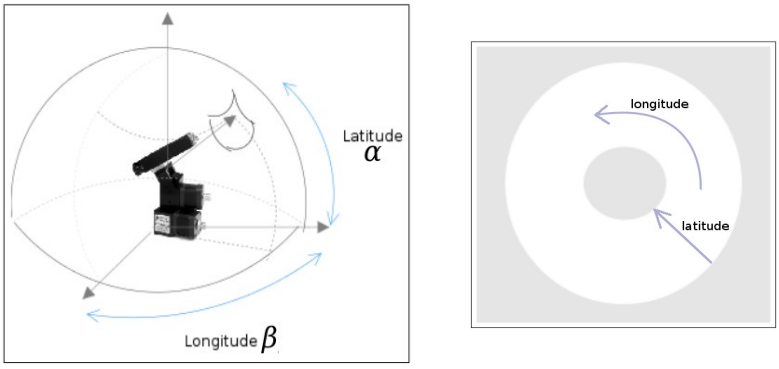
1 Sitio web: http://www.intel.com/technology/computing/opencv/

una cascada de etapas de clasificadores de características de tipo *Haar*. Cada etapa de la cascada es muy compleja y queda construida a partir de clasificadores sencillos usando votación ponderada, cuyo ajuste de pesos forma parte del propio entrenamiento. Esta configuración óptima se obtiene a través del algoritmo *AdaBoost* que, iterativamente, va seleccionando y dando pesos a los clasificadores elementales, según los propios ejemplos de entrenamiento.

Además, el detector de *OpenCV* presenta una mejora añadida, ya que, antes de aplicar el clasificador, realiza un filtro de bordes para descartar regiones demasiado uniformes donde no puede haber, por tanto, ninguna cara. Este preproceso es opcional y puede ser aprovechado para aplicar máscaras a regiones y así evitar que el detector trabaje sobre ellas.

3.2 Representación de la escena

Partiendo de la detección de caras de la imagen percibida por la cámara, hemos ampliado el campo de visión de ésta, montándola sobre un cuello mecánico que permite moverla a voluntad. Así, irá captando imágenes cuando el cuello se posa en algún punto. Este movimiento del cuello mecánico es en horizontal (*pan*) y en vertical (*tilt*), describiendo por tanto una especie de cúpula *(ver figura 3.2-a)*. Si por cada una de las posiciones que puede tomar el cuello (coordenadas *(pan, tilt)*) se tomara una imagen con la cámara de abordo, se tendría una gran imagen de escena compuesta por pequeñas imágenes monoculares *(figura 3.2-b)*.



(a) (b)

Figura 3.2: (a) Cúpula descrita mediante el movimiento del cuello mecánico; y (b) imagen resultante de la escena

Ahora bien, para formar esta imagen de escena hay que realizar una correspondencia entre los píxeles de la imagen monocular con los píxeles de la imagen de escena, ya que la imagen monocular se proyecta en esta segunda. Dependiendo de la posición del cuello mecánico, la imagen monocular obtenida por la cámara tiene coordenadas cartesianas *(u, v)* pero, por otro lado, la imagen de escena tiene como coordenadas *(α, β) (figura 3.2-a)*. Para poder construir esta imagen de escena, hay que transformar cada coordenada *(u, v)* de la imagen monocular (correspondientes a cada uno de sus píxeles) a las coordenadas *(α, β)* de la imagen de escena.

Se sabe que en la columna *u=0* de la imagen monocular tiene como valor α*: pan + ($\Delta\alpha$/2)* y el valor α en la última columna (*u=u_{max}*) de la imagen tiene como valor: *pan − ($\Delta\alpha$/2)*, siendo *pan* el valor *pan* que tiene el cuello en ese momento y $\Delta\alpha$ el valor de la apertura de la cámara en horizontal, que en el caso de la cámara *firewire* (la empleada en la plataforma final) es de 48º. Con esto se obtienen dos puntos, *P1(0, $\Delta\alpha$/2 + pan)* y *P2(u_{max}, pan − $\Delta\alpha$/2),* de la recta que transforma coordenadas de imagen en coordenadas de escena y viceversa. En ella se sustituye el valor *u* por la columna y se haya el valor α. Esta ecuación es *(3.1)*:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) \Longrightarrow \alpha = \frac{P_2.y - P_1.y}{P_2.x - P_1.x}(u - P_1.x) + P_1.y \qquad (3.1)$$

Para hallar los valores β se hace básicamente lo mismo (con *v=0* y *v=v_{max}* respectivamente). Señalar que el valor de $\Delta\beta$ se corresponderá con la apertura vertical de la cámara *firewire*, cuyo valor es 37º.

3.3 Reparto de la mirada. *Dinámica saliencia*

Una vez establecidas las coordenadas de representación en escena, es necesario controlar convenientemente el movimiento del cuello mecánico para que dirija el foco de atención hacia tal posición. Además, ante la existencia de varias caras detectadas y situadas en la memoria local de la escena, hay que tener algún tipo de mecanismo de decisión que indique al sistema dónde ha de mirar en el siguiente instante.

Para gobernar el movimiento del cuello mecánico se ha introducido la *dinámica de saliencia* y los puntos de atención. Éstos representan a las caras detectadas en la escena. Cada una de ellas contiene la posición en la escena *(α, β)* que hay que comandar para dirigir el cuello hacia ella.

Saliencia es todo aquello que llama la atención o que sobresale en una situación determinada, de ahí que el foco de atención pueda ir variando

con el paso del tiempo. En este sistema la saliencia indicará qué punto de atención ha de ser el siguiente en ser visitado. Cada cara detectada tiene una saliencia asociada, que crece con el paso del tiempo y se anula cada vez que se visita. Así, si tenemos un punto de atención con una saliencia muy alta será el próximo en ser visitado, ya que es un punto que llama la atención; si la saliencia es baja, no será visitado.

Una forma de decidir la saliencia que posee cada punto de atención es en función del tiempo que hace que no se visita. Cuando un punto se visita, su saliencia se pone a 0. Por el contrario, un punto que hace tiempo que no se ha visitado llamará más la atención que uno que se ha atendido recientemente. El sistema sigue de este modo el comportamiento de un ojo humano, ya que, según estudios de biología *(Itti, 05)*, cuando el ojo responde a un estímulo que aparece en una posición que ha sido previamente atendida, el tiempo de reacción suele ser mayor que cuando el estímulo aparece en una posición nueva. Este efecto se conoce como *inhibición de retorno* (al lugar).

El algoritmo diseñado permite que el sistema alterne el foco de atención de la cámara entre las diferentes caras existentes en la escena según la saliencia de éstas. En nuestro sistema, hemos considerado que todas las caras tienen la misma preferencia de atención, por lo que todas son observadas durante el mismo tiempo y con la misma frecuencia. Si quisiéramos asignar diferentes prioridades a las caras, podríamos establecer distintas tasas de crecimiento de la saliencia. Este hecho provocaría que el cuello se posara más veces en aquellas caras cuya saliencia crece más deprisa.

El problema de cómo revisitar un punto lo hemos abordado desde la interpretación espacial, ya que lo consideramos como un problema de evolución de las hipótesis en el lapso de tiempo entre las detecciones en t y en $t+n$ (siendo n el tiempo que llevamos sin revisitarlo). Hemos supuesto que una cara detectada volverá a ser hallada en las cercanías de donde estuvo previamente.

3.4 Seguimiento con movimiento

Cuando el sistema de reparto de mirada elige una cara, la va a estar mirando durante un cierto tiempo (3 segundos); incluso siguiéndola espacialmente si la cara se mueve. Para este seguimiento, con objeto de evitar excesivas oscilaciones y tener un control más preciso sobre el cuello mecánico, hemos decidido implementar un controlador *P* para controlar la velocidad en *pan* y *tilt* del mismo y de este modo centrar continuamente la cara

objetivo en la imagen. Este controlador *P* o proporcional permite comandar velocidades elevadas al cuello si el foco de atención al que debe dirigirse está muy alejado de la posición actual, o velocidades bajas si se precisa de pequeñas correcciones. A continuación podemos ver en la *figura 3.3* la gráfica de salida del controlador *P* ante distintas posiciones del punto de interés.

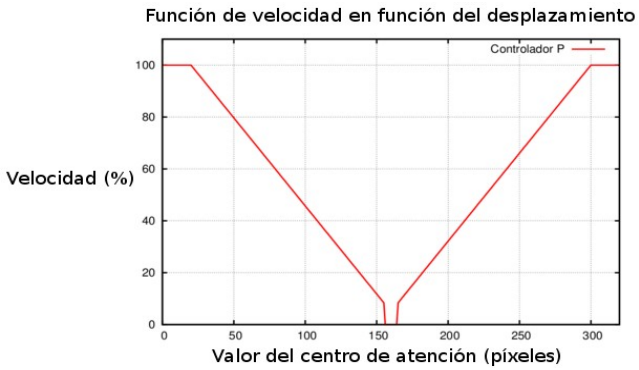


Figura 3.3: Función de velocidad del cuello según desplazamiento

3.5 Exploración de nuevas caras

En cualquier momento, y de modo sistemático, puede interesar la búsqueda de nuevas caras en la escena. Para ello se insertan periódicamente (cada *tiempoBusquedaForzada*) puntos de exploración, o *caras virtuales* con alta saliencia en la memoria local. Esta búsqueda puede interesar, sobretodo, al principio de la ejecución, momento en el que aún se desconocen las zonas de la escena en donde hay caras a seguir.

Los puntos de exploración pueden ser de dos tipos: aleatorios y de recorrido. La generación de los primeros consiste en ir asignando unas coordenadas *(pan, tilt)* de forma completamente aleatoria, dentro del rango de recorrido del cuello mecánico *(pan = [-159, + 159], tilt = [-31, +31])*. Los de recorrido nos asegurarán que en el transcurso de la ejecución todas las zonas de la escena serán supervisadas. Así, estos puntos irán desde la posición más baja de *pan* a la posición más alta, e igualmente con la coordenada *tilt*.

Los puntos de atención, sea cual sea su tipo, tendrán una saliencia inicial alta para que sean visitados más rápidamente y de ese modo comprobar si en ellos existe alguna cara. Si fuera el caso, estos puntos o caras virtuales seguirán existiendo, ya que pasarán a ser caras reales y se tratarán como tales. Ésta es la manera en la que las caras nuevas entran en el siste-

ma: se insertan en la memoria de caras y entran en la dinámica de reparto de la mirada.

Habrá una gran proliferación de caras virtuales al principio, ya que es en ese momento cuando más interesa buscar caras en la escena; puesto que partimos del absoluto desconocimiento del entorno. A medida que vayamos descubriendo caras, el afán por explorar nuevas zonas irá disminuyendo de forma proporcional al número de éstas según la ecuación *(3.2)*:

*tiempoBusquedaForzada = contadorCaras * TIEMPO_BUSQUEDA (3.2)*

3.6 Representación interna del entorno. *Dinámica vida*

Como ya se ha comentado en apartados anteriores, nuestro sistema de atención visual estará siempre guiado por el seguimiento de caras dentro de la escena. Puede atender a varias personas que haya ido detectando con el tiempo y almacenando en memoria local, alternando entre ellas, aunque no estén dentro del campo de visión inmediato de la cámara. Las personas se mueven y eventualmente desaparecen de la escena, con lo que deben ser eliminadas del sistema para mantener la representación de la escena coherente con la realidad.

Para cumplir esta labor de olvido de antiguas caras se ha implementado la dinámica denominada como *vida*. Con este mecanismo se puede saber si un objeto ha salido de la escena o si aún sigue en ella. Su funcionamiento es inverso al de la saliencia; esto es, un objeto frecuentemente visitado tendrá mayor vida que uno que apenas se visita. Si la vida de un objeto es inferior a un determinado umbral, éste se descartará y no se volverá a visitar.

Para implementar esta dinámica cada vez que se visita un objeto su vida se incrementa un poco, con un límite máximo para evitar saturación. La vida de los objetos no observados irá disminuyendo con el paso del tiempo. Así, si la vida de un objeto es superior a un cierto umbral, es que todavía sigue en la escena; en cambio, si está por debajo es que ha desaparecido.

4 Implementación software

El mecanismo de atención descrito ha sido implementado sobre el marco de la arquitectura software *JdeRobot*[2] *(Cañas, 03)*. En esta plataforma, el comportamiento se rige según la acción conjunta entre percepción y actua-

2 Sitio web: http://jde.gsyc.es/index.php/Main_Page

ción. Ambos están divididos en pequeños componentes llamados esquemas, organizados en una jerarquía dinámica.

Nuestro mecanismo de atención se ha acoplado como un esquema más de percepción (*iFollowFace*) de esta arquitectura *(ver figura 4.1)*, que se encarga de construir y anclar una representación de la escena. Esta representación se compone de un conjunto de caras humanas pertenecientes al entorno del robot. Como la cámara sola no cubre todo el espacio alrededor del robot, no todas las posibles caras de la escena pueden ser detectadas en un momento determinado por la cámara. Este esquema perceptivo conlleva, por tanto, una percepción activa *(Bajcsy, 88)* que mueve la unidad *pantilt* a fin de buscar objetos y mantener su representación interna constantemente actualizada. Dicho de otro modo, la regla aquí es "actuar para percibir"; en contraposición a "percibir para actuar".

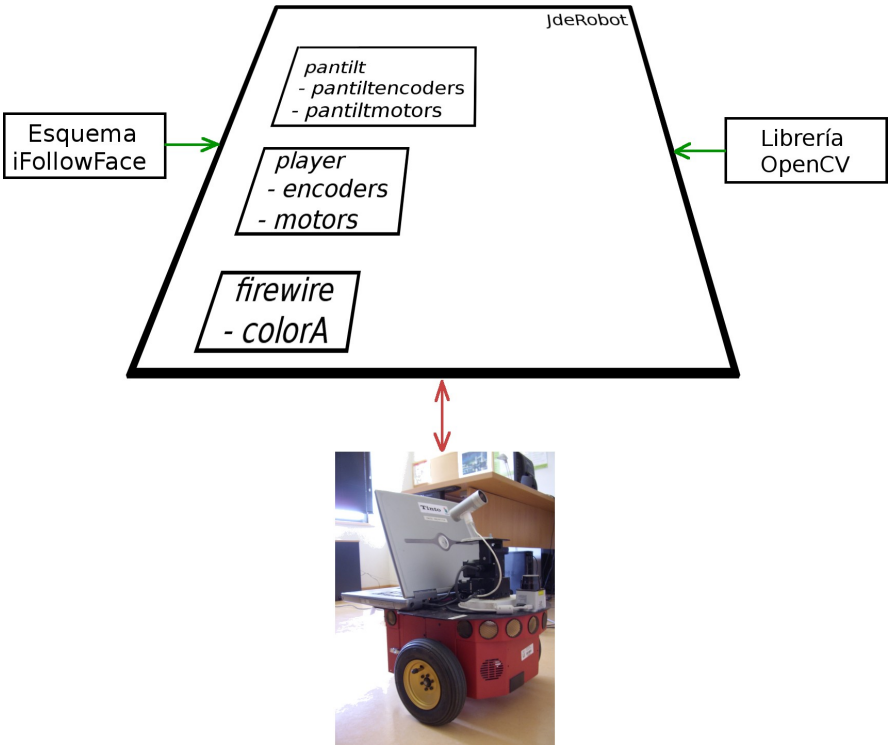


Figura 4.1: Arquitectura del sistema

Los objetos en el entorno del robot guían los movimientos de la cámara, de modo que el mecanismo de atención es de abajo hacia arriba (*bottom-up*). El único mecanismo de arriba hacia abajo (*top-down*) existente es que

los objetos relevantes son aquéllos que tienen apariencia de cara humana. Esta tendencia a mirar hacia las caras es similar a la predisposición detectada por etólogos en los animales hacia determinados estímulos, según en qué contexto *(Tinbergen, 51)*.

El sistema de atención visual aquí presentado se ha implementado siguiendo un diseño de máquina de estados, que determina cuándo se ejecutan los pasos descritos en la *sección 3*. Así, podemos distinguir cuatro estados:

- Deliberar próximo objetivo (estado 0)
- Completar movimiento sacádico (estado 1)
- Analizar imagen para encontrar posibles caras (estado 2)
- Seguir cara detectada (estado 3)

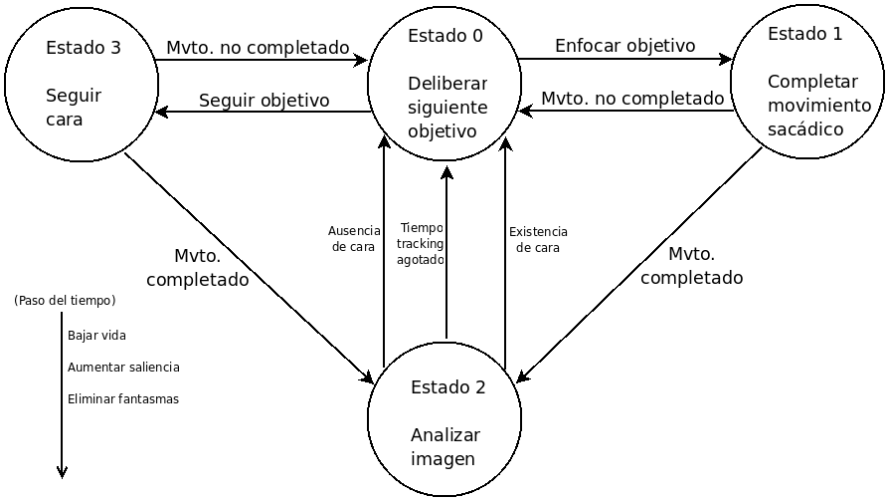


Figura 4.2: Diagrama de estados del sistema

El funcionamiento es el reflejado en la *tabla 4.3*. Inicialmente lo que hacemos, con el paso del tiempo, es ir actualizando las posibles caras que tengamos ya almacenadas en memoria. Por un lado comprobar si alguna cara ya está desfasada, porque su vida es inferior a un determinado umbral, y por otro, aumentar la saliencia y disminuir la vida.

Partiendo del estado inicial o estado 0, el sistema se pregunta si hay algún objetivo a mirar (por si tenemos alguna cara previamente almacenada en memoria); en caso afirmativo, pasaremos al estado 1. En caso negativo, crearemos una cara virtual, e insertaremos en memoria. Vuelta al estado 0.

En el estado 1 el cometido es completar el movimiento hasta llegar a la posición absoluta indicada por el estado 0. Una vez ahí, pasaremos al esta-

do 2 donde analizaremos si hay caras o no. En cualquier caso, de éste pasamos al estado 0 y vuelta a empezar.

Del estado 0 sólo pasaremos al 3 si en el último objetivo marcado se ha encontrado alguna cara, en cuyo caso se podrá hacer el seguimiento de la misma. Éste es precisamente el propósito de dicho estado.

Tabla 4.3: Pseudocódigo del comportamiento del sistema

```
actualizarCaras (instanteActual)
tiempoDeBusquedaForzada = numeroCaras x TIEMPO_BUSQUEDA_FORZADA

si ((estado = 0) Y ((contadorCaras = 0) O ((noEstaVacia(listaCaras)) Y
        ((instanteActual - instanteUltimoForzado) > tiempoDeBusquedaForzada)))) entonces
        si ((noEstaVacia(listaCaras)) Y
        ((instanteActual - instanteUltimoForzado) > tiempoDeBusquedaForzada)) entonces
                busquedaForzada = CIERTO
                instanteUltimoForzado = instanteActual
        fin si
        generarCaraVirtual (instanteActual)
fin si

sino si (((estado = 0) Y ((contadorCaras > 0) Y ((detectarCaras() = FALSO) O
        (detectarCaras() <= 0)))) O (busquedaForzada = CIERTO)) entonces
        estado = 1;
        busquedaForzada = FALSO;
fin sino

sino si ((estado = 0) Y (contadorCaras > 0) Y (detectarCaras() > 0)) entonces
        estado = 3;
fin sino

sino si ((estado = 1) && (movimientoCompletado)) entonces
        estado = 2;
        movimientoCompletado = FALSO;
fin sino

sino si (estado = 2) entonces
        estado = 0;
fin sino

sino si ((estado = 3) && (movimientoCompletado)) entonces
        estado = 2;
        movimientoCompletado = FALSO;
fin sino
```

5 Experimentos

Nuestros experimentos[3] han sido realizados con un robot real *Pioneer 2DX* (de *ActivMedia Robotics*), sobre el cual se ha montado un ordenador portátil *Dell*, con procesador *Intel Centrino* a 1.7 Ghz. y bajo el S.O. *Linux Ubuntu 8.04* (*hardy*). Además se le ha instalado un cuello mecánico (*Pan-tilt Unit 46-17.5* de *Directed Perception*) con libertad de movimiento [+180º, -180º] en *pan* y [+31, - 80] en *tilt;* capaz de desarrollar una velocidad mínima de 0.0123º/seg. y máxima de 300º/seg. en ambos ejes. A su vez, en éste se ha colocado una cámara *firewire iSight* (de *Apple*) con *autofocus* y apertura focal 60º y 40º en horizontal y vertical respectivamente. La alimentación eléctrica de la unidad *pan-tilt* es suministrada por la base del robot; y las órdenes comandadas al mismo se realizan a través del puerto serie.

En estas pruebas hemos dividido nuestros intereses simplemente en analizar el comportamiento del sistema según las caras[4] que haya en la escena. En el primer experimento probamos la función de seguimiento con una única cara, mientras que en el segundo analizamos el reparto de mirada (saliencia) y el olvido de antiguas caras (vida).

5.1 Seguimiento de una cara

En este primer experimento partimos con 0 caras detectadas (como ocurrirá siempre); así, el sistema tiene que comandar al cuello mecánico que realice movimientos sacádicos en busca de caras humanas por toda la escena. Estos movimientos son cortos, precisos y rápidos; lo justo para que de tiempo a examinar si hay o no cara en la imagen actual recibida con la cámara. Transcurrido un cierto tiempo, el sistema detecta una cara, la única que hay presente en el entorno.

Con una única cara *(ver figura 5.1)* el seguimiento se hace con movimientos suaves, según el funcionamiento del controlador *P* comentado en la *sección 3.4*. Si hacemos movimientos bruscos alejándonos del centroide de la imagen percibida por el sistema, el movimiento del cuello será rápido; cuando nos movemos lentamente el movimiento será más parsimonio-

3 Toda la documentación, con vídeos reales e imágenes, está disponible en la web del proyecto *RobotVision*, del Grupo de Robótica de la Universidad Rey Juan Carlos: http://jde.gsyc.es/index.php/RobotVision

4 Se han empleado en ciertos momentos caras fotografiadas para facilitar las pruebas. No obstante éstas tenían que ser de apariencia real para que el sistema las identificara como tales, quedando claro que no admite caricaturas o *falsas* caras humanas.

so. Así, en los tres fotogramas de la *figura 5.1* el cuello mecánico siempre aparece alineado con la cara que está siguiendo.

Por otro lado, y como ya se ha comentado en secciones anteriores, cuando transcurre el tiempo de exploración forzada (5 seg.), el cuello mecánico realiza un movimiento sacádico para obtener una instantánea de alguna zona de la escena; al no encontrar ninguna cara, vuelve inmediatamente a seguir la única cara que ya tenía visualizada. Este tiempo no se ve incrementado en este caso, dado que el sistema sólo tiene detectada una cara. Por tanto el proceso descrito de búsqueda de nuevas caras combinado con el seguimiento de la cara detectada se repite con el tiempo.

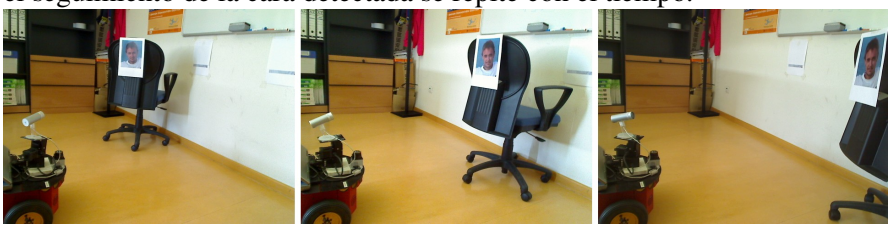


Figura 5.1: Seguimiento de una sola cara

5.1 Seguimiento de varias caras

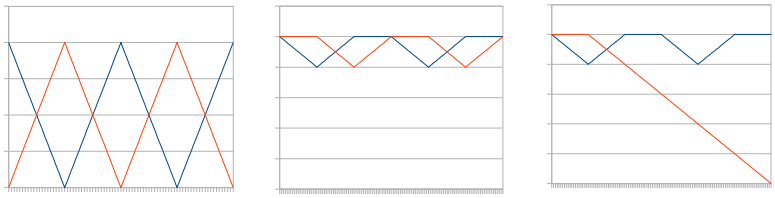
En el segundo experimento, partimos igualmente de no tener caras detectadas. Cuando detecta la primera cara *(ver figura 5.2, ap. 1)*, el tiempo de exploración forzada es de 5 segundos, tras los cuales el sistema realizará una exploración forzada por la escena. Este proceso se repite durante cierto tiempo, hasta que encuentra una segunda cara *(ver figura 5.2, ap. 2)*. Es ahora cuando el sistema puede seguir con la mirada a ambas caras detectadas. Además, como ya hay dos caras en memoria, duplicamos igualmente el tiempo de exploración forzada a 10 segundos. Este hecho nos permite mantener la mirada durante más tiempo a las caras que tenemos detectadas, así como seguir buscando otras posibles. Gracias a este mecanismo, podemos encontrar todas las caras existentes en la escena; de modo que transcurrido un cierto tiempo, el sistema encuentra la tercera y última cara presente en este experimento *(ver figura 5.2, ap. 3)*. Así que triplicamos el tiempo de exploración forzada a 15 segundos. Si nos fijamos, lo que conseguimos con este incremento de tiempo es que a medida que vamos detectando más y más caras, la búsqueda de nuevas se hará cada vez menos frecuente. No obstante, cuando toque hacer una exploración forzada, iremos igualmente a explorar nuevas zonas *(ver figura 5.2, ap. 4)*, volviendo después a la última cara en la que se quedó.

Figura 5.2: Seguimiento de varias caras

Por último, hemos reflejado en la siguiente gráfica el modo de actuar de las dos dinámicas concurrentes ya explicadas anteriormente: saliencia y vida. Corresponden al momento en que el sistema tiene dos caras detectadas (*figura 5.2, ap. 1 y 2)*. En tal situación, podemos observar en la *figura 5.3-a* cómo evoluciona la saliencia de ambas. Cuando el sistema está siguiendo una cara (color azul) su saliencia disminuye, mientras que a la otra cara almacenada en memoria (color rojo) le va aumentando hasta ganar la competición y forzar a que el sistema la *mire* a ella.

La evolución de la vida de ambas caras, cuando ambas permanecen en la escena, se aprecia en la *figura 5.3-b*. Su funcionamiento es inverso a la saliencia; esto es, cada vez que el sistema visita una cara, su vida se incrementa un poco, con un límite máximo para evitar saturación.

En la *figura 5.3*-c se refleja una situación en la que hemos ocluido una cara, con lo que el sistema deja de detectarla y, por tanto, su vida comienza a descender. Cuando su valor es inferior a un determinado umbral, esa cara se descarta y no se vuelve a visitar.



(a) *(b)* *(c)*

Figura 5.3: Gráfica de tiempos de la evolución del sistema. (a) Saliencia; (b) Vida; y (c) Vida ante la desaparición de una cara

6 Conclusiones

En este trabajo hemos presentado un sistema de atención visual global cuyo propósito es encontrar caras humanas por la escena circundante a éste, siguiéndolas con la mirada en tal caso. Para ello se ha desarrollado un mecanismo de dinámica concurrente entre vida y saliencia, en el cual la cara con mayor saliencia es la siguiente en ser visitada y, por tanto, la que dirige el movimiento del *pan-tilt* en todo momento. Así hemos conseguido que el robot siga con la mirada a todos sus interlocutores, de forma que la interacción *persona-robot* es más natural. Y la dinámica vida nos ha permitido tener una representación coherente de las caras en escena, evitando de este modo que el robot preste atención a personas que han dejado de estar allí.

Además, y dado que la escena es mayor que el campo de visión inmediato de la cámara del robot, hemos implementado una memoria local de corto plazo. Esto ha facilitado la interacción entre el robot y las personas que lo rodean, ya que éstas pueden estar situadas en posiciones que el robot no es capaz de ver en un momento dado pero en las que *sabe* que existen rostros humanos.

Los diferentes experimentos llevados a cabo nos demuestran que los comportamientos generados siguiendo este mecanismo resultan ser bastante similares a cómo presta atención un humano. Por ejemplo, ante la existencia de una sola cara en la escena, el sistema presta toda la atención a ésta, siguiéndola con la mirada, mientras que busca otras cada cierto tiempo. Por otro lado, cuando tenemos varias irá alternando entre todas ellas, habiéndose incrementado proporcionalmente el tiempo de exploración. En el lado opuesto en que no haya caras, este tiempo es mínimo; emulando así el *afán* por detectar caras.

Otro aspecto a destacar es el olvido de caras que han desaparecido de la escena, evitando así tener *fantasmas* en la memoria representativa del entorno. No obstante, han de transcurrir varios intentos fallidos para considerar la desaparición de una cara, ya que en ocasiones es posible que no se detecte por oclusiones esporádicas. Aunque el algoritmo de detección presentado suele ser bastante robusto ante diferentes condiciones de iluminación y distancia *persona-robot*.

Una de las posibles mejoras a este trabajo puede ser el reconocimiento ya no sólo de una cara humana, sino también de expresiones así como su posterior correspondencia con las emociones. Este proceso es complejo, ya que a parte de detectar la cara hay que detectar los elementos que influyen

en la formación de las distintas expresiones faciales, y posteriormente identificar la emociones correspondientes a tales expresiones.

Por otro lado, se podría emplear la atención visual para que el robot aparte de reconocer caras humanas, navegue de forma autónoma. Podría mover la cámara para detectar caras, marcas visuales y obstáculos potenciales en su entorno, como pueden ser las paredes. Una vez detectadas, el sistema las incluiría en su representación interna del mundo, y repartiría la mirada entre las caras y los obstáculos inmediatos a la navegación, lo que le podría ser útil para navegar de manera segura por él, aparte de interactuar con personas.

Referencias

Ballard, D.H. "Animate vision", in Artificial Intelligence 48, pp. 57-86, 1991.

Bajcsy, R. 1988. Active Perception. Proc. of the IEEE 76, pp. 996-1005.

Cañas, J.M., "Jerarquía dinámica de esquemas para la generación de comportamiento autónomo", PhD, Universidad Politécnica de Madrid, 2003.

Itti, L., Koch, C., "Computational Modelling of Visual Attention", in Nature Reviews Neuroscience 2, pp. 194-203, 2001.

Itti, L., Koch., C. 2005. A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Lang, S., Kleinehagenbrock, M. et. al. 2003. Providing the Basis for Human-Robot-Interaction: A Multi-Modal Attention System for a Mobile Robot. Bielefeld University, Faculty of Technology, Bielefeld, Germany.

Lienhart, R., Maydt., J. 2002. An extended set of haar like features for rapid object detection. In IEEE ICIP 2002, volume 1, pp. 900-903.

Marocco, D., Floreano, D. 2002. Active vision and feature selection in evolutionary behavioral systems. In Proc. of Int. Conf. on Simulation of Adaptive Behavior (SAB-7), pp. 247-255.

Marr, D., "Vision, A Computational Investigation into the Human Representation & Processing of Visual Information", in Freeman, San Francisco, CA., 1982.

Pylyshyn, Z., "Visual Indexes, preconceptual object and situated vision", in Cognition 80, pp. 127-158, 2001.

Tinbergen, N., "The study of instinct", in Clarendon University Press, Oxford UK, 1951.

Tsotsos, J.K., et.al., "Modeling visual attention via selective tuning", in Artifical Intelligence 78, pp. 507-545, 1995.

Viola, P., Jones., M., "Rapid object detection using a boosted cascade of simple features", 2001.

Yang, M.-H., Kriegman, D. J., Ahuja. N. 2002. Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1), pp. 34-58.

Yoshikawa, Y. et. al. 2006. Responsive robot gaze to interaction partner. Intelligent Robotics and Communication Laboratories. Advanced Telecommunication Research Institute International. Keihanna Science City, Kyoto, 619-0288 Japan.

Zaharescu, A., Rothenstein, A.L., Tsotsos, J.K. 2005. Towards a biologically plausible active visual search model. In Proc. of Int. Workshop on Attention and Performance in Computational Vision, WAPCV-2004, Springer LNCS 3368, pp. 133-147.