CAPÍTULO 7

MEMORIA VISUAL ATENTIVA BASADA EN CONCEPTOS PARA UN ROBOT MÓVIL

J. VEGA[1], J.M. CAÑAS[1], P. MIANGOLARRA[1] y E. PERDICES[1]

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

Resumen. Los sistemas de visión son hoy en día uno de los elementos sensoriales más utilizados en robótica autónoma. Su principal dificultad radica en extraer información útil de las imágenes capturadas, así como el pequeño campo visual de las cámaras convencionales.

Sin embargo, con *cámaras activas* es posible revisitar características de un área previamente visitada, incluso si tal zona está fuera del alcance visual inmediato. Para tener información precisa sobre las zonas de interés que rodean al robot es necesario un *mapa de memoria* detallado del entorno. Dado que el coste computacional de mantener tal cantidad de información es elevado, podemos mantener sólo unas pocas referencias.

Aquí presentamos una serie de trabajos preliminares de un mecanismo de *memoria visual atentiva*, validándolos experimentalmente sobre un robot móvil real. Mientras que en la memoria se mantiene información sobre distintos objetos de interés que el robot va encontrando por el entorno por el que navega; por su parte, los algoritmos de atención desarrollados se encargan de mantener esta memoria sincronizada con el entorno real: revisitando los objetos ya almacenados en ella, siguiéndolos con la mirada, o buscando otros nuevos.

Palabras clave. Atención visual, reconocimiento de objetos, exploración de escena, seguimiento visual, memoria visual.

1 Introducción

Aunque la visión computacional no ha sido la modalidad sensorial más empleada hasta hace unos años en robótica móvil (sónar y/o láser han sido mayoritariamente usados como sensores), actualmente se ha convertido en el sensor que más se está usando y -sin duda- se usará con mayor profusión a largo plazo, debido a las posibilidades que ofrece. Son dispositivos de bajo coste y potencialmente muy ricos, ya que ofrecen mucha información.

Mientras que la investigación a nivel internacional en visión computacional está creciendo muy rápidamente, la atención visual para tareas de robótica no resulta una técnica fácil. La principal dificultad radica en extraer información útil del gran caudal de datos que vierte una cámara. Así, los propósitos de esta rama de estudio se está ampliando a otras muchas aplicaciones; desde reconocimiento de caras y sistemas de videovigilancia, hasta la adquisición de modelos tridimensionales para entornos de realidad virtual.

La visión es el sensor cuya habilidad principal reside en dar información sobre *qué* y *dónde* se encuentran los objetos que el robot va encontrando a su paso. Y, aunque debemos ser cautos a la hora de comparar un robot con un organismo biológico *(Nehmzow, 93)*, lo que sí está claro es que la vista es el sentido principal en que se apoyan los animales para moverse por el entorno.

Los humanos disponemos de un preciso sistema de visión activa *(Bajcsy, 88)*. Esto significa que podemos concentrarnos en determinadas regiones de interés de la escena que nos rodea, gracias al movimiento de los ojos y/o de la cabeza *(Vega, 09)*, o simplemente repartiendo la mirada en distintas zonas dentro de la imagen actual que estemos percibiendo. Y ¿qué ventajas nos ofrece esta solución natural frente a una mecanismo de atención pasivo donde los sensores se centran por igual en todas las zonas de la imagen?

1. Zonas de la escena que pueden ser no accesibles por sensores fijos, sí lo son por sensores móviles.
2. Dirigiendo la atención a zonas específicas de la imagen que son interesantes podemos evitar costes en visitar zonas que no deseamos. Por ejemplo, en la tarea de coger algo, los humanos nos concentramos solamente en mover tal objeto.

Así, podemos considerar la visión activa como un supervisor de un amplio repertorio de tareas, notablemente mayor que el que contempla la visión pasiva.

En este artículo mostramos un sistema de atención visual global (*overt*) implantado sobre el robot real *Pioneer*, equipado con un cuello mecánico en el que hay montada una sola cámara *firewire*. Este sistema permite al robot detectar distintos objetos situados en el espacio que lo rodea, como flechas, objetos con formas cuadriculadas y otras formas básicas; así como rostros humanos. Para ir probando todas las prestaciones de los algoritmos desarrollados, simplificando la geometría del modelo matemático empleado, vamos a suponer en todo momento que los objetos están apoyados sobre el suelo.

Reconocimiento que le será muy útil para tomar decisiones inteligentes. Estos objetos entran en una doble dinámica de *vida/saliencia* que guía el comportamiento del sistema en todo momento. De este modo el robot es capaz de saber dónde están los objetos y de qué objetos se tratan.

2 Estado del arte. Atención visual en robots

La atención visual 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, 08)*.

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 am-

plio. 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)*.

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

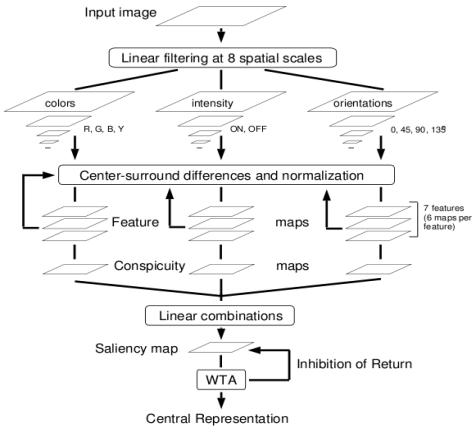

Figura 2.1: Esquema de formación del mapa de saliencia

3 Diseño general

El objetivo general de nuestro sistema es realizar un seguimiento de atención sobre los distintos objetos básicos presentes en la escena circundante al robot. Por tanto, se deben captar nuevos objetos, repartir la mirada sobre los allí existentes y eliminarlos de la memoria una vez hayan desaparecido. Para mantener esta información coherentemente con la realidad necesitamos de mecanismos que refresquen convenientemente una memoria 3D a corto plazo. El diseño general podemos verlo en la *figura 3.1*.

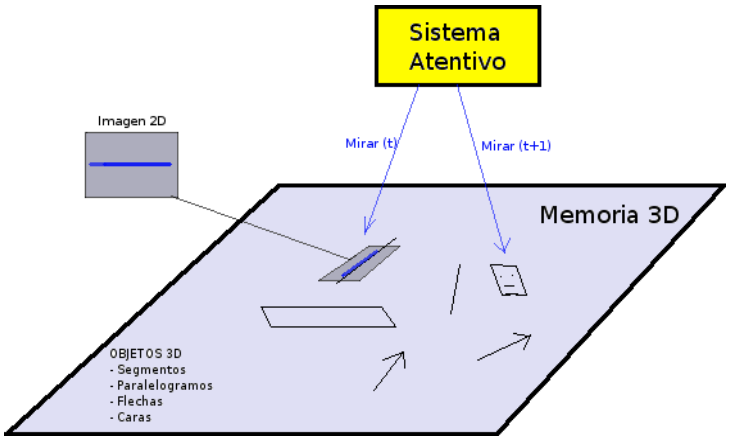

Figura 3.1: Diagrama del sistema perceptivo visual implementado

El sistema atentivo es el encargado de indicar al robot cuál es la siguiente zona a la que dirigir la mirada, que puede ser una zona ya explorada a revisitar, una nueva zona de exploración, o bien una zona no explorada de corroboración de hipótesis. Además, también se encarga de centrar la mirada sobre el objeto al que se está prestando atención en un determinado momento, siguiéndolo en caso de que éste se mueva.

Por su parte, en el mecanismo de memoria 3D la primera fase es el análisis 2D, que detecta segmentos 2D presentes en la imagen actual, así como rasgos característicos de una cara humana. A continuación, el algoritmo de reconstrucción 3D sitúa estos objetos en el espacio 3D, bajo la *hipótesis suelo*; esto es, consideramos que todos los objetos están apoyados sobre el suelo. Y finalmente la memoria 3D almacena la posición de éstos en el espacio 3D, genera hipótesis perceptivas y calcula predicciones de estos objetos en la imagen en curso que está percibiendo el robot.

4 Memoria visual 3D

En esta sección veremos los distintos componentes en los que se basa el sistema de memoria implementado para el sistema de atención. Por un lado el *detector de objetos*, que es el encargado de identificar las formas básicas así como las caras humanas que hay en la imagen en curso. Por su parte, el mecanismo de *predicción* de elementos permitirá al sistema predecir elementos ya memorizados con anterioridad, aliviando el coste computacional. Y también resaltaremos el algoritmo de generación de *hipótesis per-*

ceptivas sobre los elementos almacenados, que permitirá al sistema abstraer objetos complejos. Y todo ello apoyado sobre la *memoria visual*, que conforma el núcleo de todo el sistema atentivo. Ésta permite ampliar el campo de visión a toda la escena circundante al robot; no sólo el campo visual instantáneo.

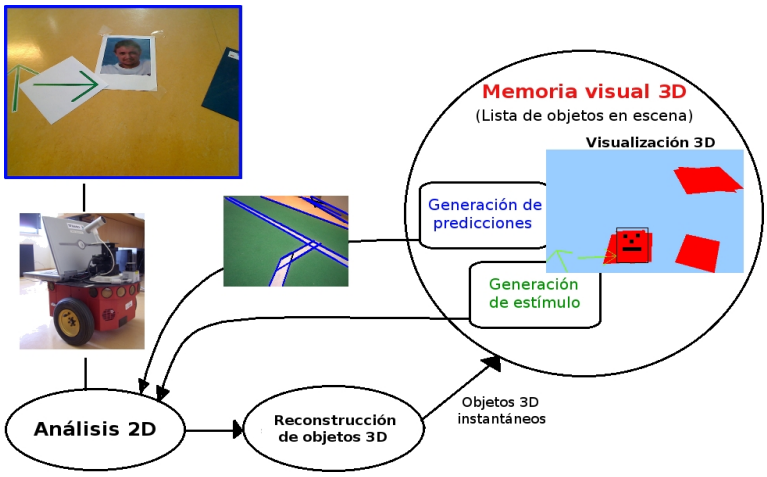

Figura 4.1: Diagrama de bloques del sistema de memoria visual

4.1 Procesamiento imágenes 2D

El principal objetivo de esta parte del sistema es extraer como primitiva básica el segmento recto en 2D, y rasgos humanos (en el caso de las caras). Estas primitivas son las que manejará el reconstructor 3D.

El sistema de detección 2D a su vez está conectado a la memoria 3D directamente, para así ahorrar tiempo de cómputo de reconstrucción de aquellos objetos que ya pueda haber almacenados en memoria; además, nos sirve para corroborar/refutar los objetos instantáneos con los memorizados. Asimismo la imagen actual nos sirve para confirmar estructuras previamente visualizadas de forma parcial.

El primer paso para simplificar la imagen es un filtrado de bordes mediante el *Algoritmo de Canny*. Posteriormente aplicamos la *Transformada de Hough* para extraer solamente segmentos rectos. En el caso de detección de caras humanas empleamos el detector basado en *Filtros Haar* en cascada *AdaBoost* (*Viola y Jones, 01)* y *(Lienhart y Maydt, 02)*. Para realizar estos procesamientos nos apoyamos en la librería *OpenCV*.

En la *figura 4.2* vemos la reconstrucción de segmentos 3D antes y después del postprocesado de Hough.

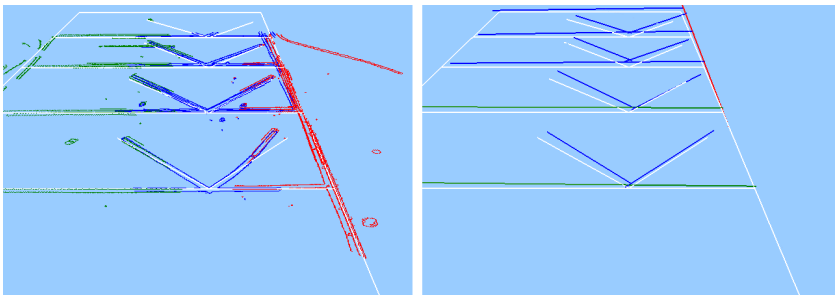

Fig. 4.2: Reconstrucción segmentos 3D antes y después del postprocesado

Antes de extraer características de la imagen en curso, el sistema hace la predicción de aquellos objetos almacenados en la memoria 3D que deberían ser visibles desde la posición actual. Para cada objeto 3D almacenado y que debería ser visible hacemos su proyección sobre el plano imagen.

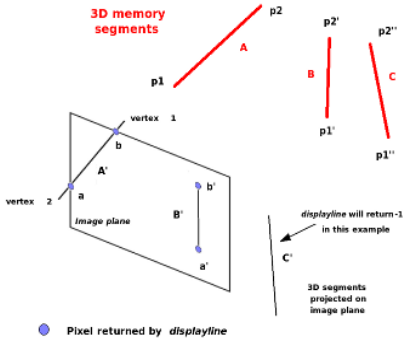

Figura 4.3: Proyección de segmentos 3D sobre el plano imagen

Acto seguido refutamos/corroboramos tales segmentos predichos; comparamos uno por uno de estos segmentos, con los obtenidos mediante la Transformada de Hough. Esta comparación da lugar a tres conjuntos de segmentos, como podemos ver en la *figura 4.4*.

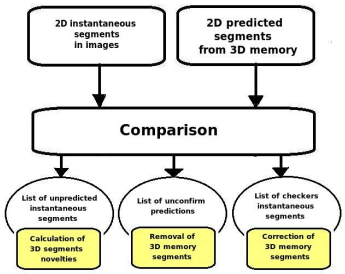

Figura 4.4: Emparejamiento entre segmentos predichos e instantáneos

4.2 Reconstrucción instantánea con segmentos 3D

El mecanismo anterior nos vierte un conjunto de segmentos 2D que ha de ser situado en el espacio 3D. Para ello nos basamos en la idea de *Hipótesis suelo*. Dado que contamos con una sóla cámara, necesitamos de alguna restricción que nos posibilite estimar la tercera dimensión. Suponemos que todos los objetos están apoyados en el suelo.

Una vez que tenemos los objetos 3D, y antes de incluirlos en la memoria 3D, es necesario hacer un postprocesado para evitar posibles duplicados en memoria debido a ruido en las imágenes. En este postprocesado comparamos la posición relativa entre segmentos, así como su orientación y proximidad y, en caso de que estén repetidos, se fusionan. La salida es un conjunto de segmentos 3D relativos al sistema de coordenadas del robot. La *figura 4.5* muestra la escena 3D con los objetos reconstruidos por parte del sistema, así como los segmentos detectados en la imagen actual y los segmentos predichos desde tal posición.

Empleamos en total cuatro sistemas de coordenadas para definir el modelo geométrico:

- El sistema de coordenadas absoluto, cuyo origen está en algún punto del mundo por el que se mueve el robot.
- El sistema situado en la base del robot. La odometría del robot nos da su posición y orientación, con respecto al anterior sistema.
- El sistema de la base del cuello mecánico. Tiene sus propios *encoders* para conocer su posición en un momento dado, con movimientos de *pan* y *tilt* con respecto a la base del robot.
- Y por último tenemos el sistema de coordenadas de la propia cámara, desplazada y orientada en un determinado eje respecto al cuello mecánico.

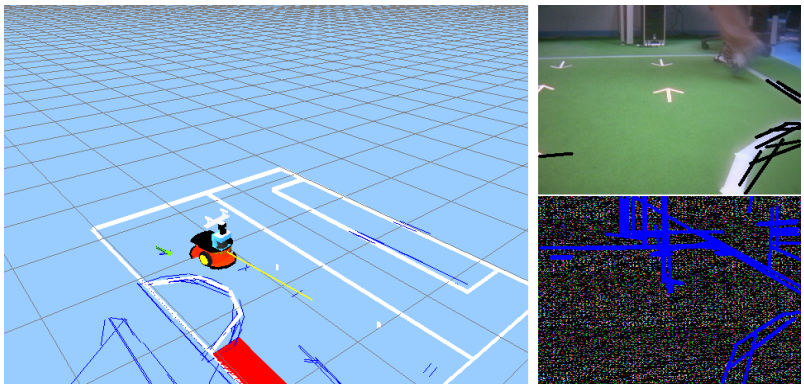

Figura 4.5: Reconstrucción de escena 3D, seg. predichos e instantáneos

En la *figura 4.6* podemos ver una ilustración de estos sistemas de referencia.

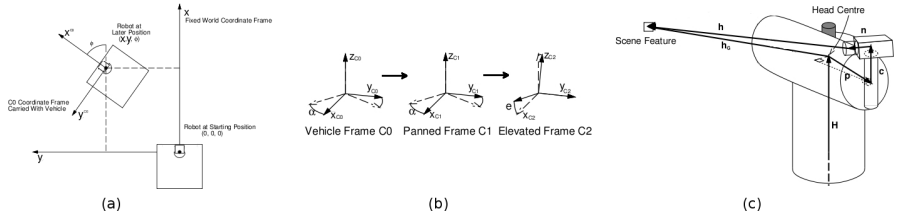

Fig. 4.6: Sist. de coordenadas. (a)Base robot; (b)pan y tilt; y (c)cámara

4.3 Inserción de segmentos en memoria, borrado y corrección

La memoria 3D está formada por un conjunto de listas dinámicas en las que almacenamos información sobre los distintos tipos de elementos presentes en la escena (posición, tipo, color, etc.). Partiendo de la forma más básica de estructura, el segmento, -y gracias a la memoria- podemos establecer relaciones entre ellos para dar lugar a otro tipo de elementos más complejos como flechas, objetos paralelogramos, triángulos o círculos.

El proceso de incorporación de segmentos a memoria 3D consiste básicamente en comparar uno por uno cada segmento calculado en la imagen instantánea con aquéllos ya almacenados. En caso de que haya segmentos cercanos y con una orientación similar, el sistema fusiona tales segmentos en uno nuevo tomando la mayor de las longitudes de sus predecesores, y la orientación del más reciente, ya que muy probablemente es el más coherente con la realidad (los más antiguos suelen tener mayor ruido debido a los errores odométricos del robot).

Para hacer más liviano computacionalmente este proceso de fusión, el sistema cuenta con una memoria caché compuesta únicamente por aquellos segmentos más cercanos al robot (en un radio de 4 m. a su alrededor).

Si un segmento detectado en la imagen actual no coincide con lo predicho, el sistema crea uno nuevo que posiblemente reemplazará al ya existente (reemplazo o corrección) si cumple ciertas restricciones, ya que al ser visto recientemente tiene mayor validez que uno antiguo. Para reflejar este hecho disponemos de un parámetro denominado *incertidumbre* que irá aumentando a medida que un segmento va pasando tiempo en memoria.

El proceso de borrado de elementos se basa en el mismo principio, pero aquí las restricciones son más permisivas. Así conseguimos que el proceso de reemplazo sea prioritario frente al borrado.

4.4 Percepción estructurada

Nuestro modelo de objetos consiste en un conjunto de segmentos cuyos vértices pueden pertenecer a estructuras más abstractas. Y no sólo eso, sino que son vértices etiquetados según el número de segmentos que tengan atados a ellos. Esto requiere tener un modelo de objeto para aquellos casos en que determinados vértices no sean visibles en un momento dado. Por ejemplo, para un paralelogramo cualquiera, este número mínimo de vértices visibles es pequeño; con que tengamos tres vértices somos capaces de estimar el cuarto. Este vértice pasará a ser un nuevo punto de atención, para así comprobar la hipótesis perceptiva.

Para almacenar un segmento tenemos la estructura *Segment3D*, compuesta -entre otras cosas- de punto inicial y final, así como un puntero a otro tipo de posibles estructuras de las que puede formar parte: *Arrow3D* (flecha) o *Parallelogram3D* (estructuras cuadriculadas).

Los segmentos y sus correspondientes vértices son usados para detectar paralelogramos comprobando la conexión entre ellos, así como el posible paralelismo existente. El análisis de los ángulos que forman entre sí los segmentos proporciona información acerca de cómo los segmentos están conectados unos con otros. Además, esta característica puede ser empleada para fusionar segmentos que se ven de forma incompleta o discontinua. Del mismo modo, también podemos extraer la posición de un posible cuarto vértice, haciendo uso de la información que nos dan los otros bordes y/o vectores del posible paralelogramo.

Esta capacidad da a nuestro algoritmo una gran robustez frente a oclusiones, las cuales ocurren frecuentemente en el mundo real. La *figura 4.7-b,c* ilustra un ejemplo de vistazo parcial de paralelogramos. Tras varios vistazos el algoritmo es capaz de reconstruir totalmente tales paralelogramos, como vemos en la *figura 4.7-a*.

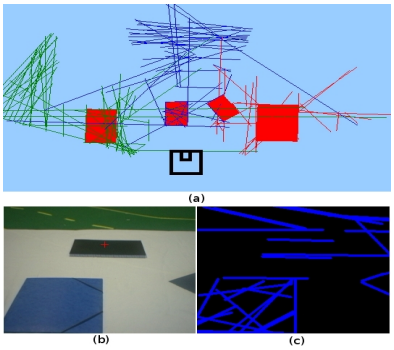

Figura 4.7: Generación de hipótesis paralelogramo, con oclusión

5 Sistema de atención visual

En el anterior apartado hemos descrito detalladamente el funcionamiento de la memoria visual donde se iban situando los objetos detectados, los cuales tenían ciertos atributos. Pues bien, ahora pasaremos a describir el mecanismo de atención visual implementado basándonos en dos de éstos atributos: *saliencia* y *vida* (ver *fig. 5.1*). Por un lado la saliencia permite decidir dónde mirar en cada momento, mientras que la vida es el mecanismo para olvidar un objeto que haya desaparecido de la escena. Tendremos así puntos de atención según la necesidad del sistema: revisitar zona para refrescar memoria, explorar, o bien comprobar una hipótesis perceptiva.

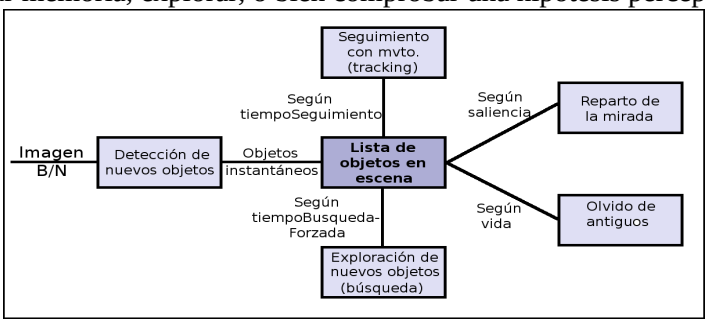

Figura 5.1: Diagrama de bloques del sistema de atención

Además, tendremos un mecanismo de *seguimiento* continuo de éstos con movimientos de la cámara, implementado como un controlador *P*; y otro mecanismo que nos permitirá *explorar* nuevas zonas desconocidas de la escena.

5.1 Reparto de la mirada. *Dinámica saliencia*

Una vez establecidas las coordenadas de representación en la escena para cualquier objeto, es necesario controlar convenientemente el movimiento del cuello mecánico para que dirija el foco de atención hacia tal posición. Además, ante la existencia de varios objetos detectados y situados 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 los objetos detectados en la escena. Cada uno de ellos contiene la posición en la

escena 3D *(X, Y, Z)*, que se traduce en comandos al cuello mecánico para dirigir el foco de atención hacia tal elemento.

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 elemento en memoria 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*.

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

Hemos supuesto que un objeto detectado volverá a ser hallado en las cercanías de donde estuvo previamente.

5.2 Seguimiento continuo con movimiento

Cuando el sistema de reparto de mirada elige un punto de atención, lo va a estar mirando durante un cierto intervalo (rodaja de tiempo); incluso siguiéndolo espacialmente si éste 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 el objetivo en la imagen.

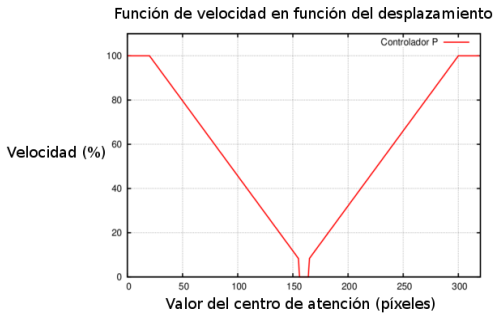

Figura 5.2: Función de velocidad del cuello según desplazamiento

Este controlador 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.

5.3 Exploración de nuevas zonas de interés

En cualquier momento, y de modo sistemático, puede interesar la búsqueda de nuevos objetos en la escena. Para ello se insertan periódicamente (cada *tiempoBusquedaForzada*) puntos de exploración 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 objetos de interés.

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 algún objeto de interés. Si fuera el caso, estos puntos seguirán existiendo. Ésta es la manera en la que los nuevos objetos entran en el sistema: se insertan en la memoria y entran en la dinámica de reparto de la mirada.

Habrá una gran proliferación de puntos de exploración al principio, ya que es en ese momento cuando más interesa buscar zonas de interés en la

escena; puesto que partimos del absoluto desconocimiento del entorno. A medida que vayamos descubriendo objetos, el afán por explorar nuevas zonas irá disminuyendo de forma proporcional al número de éstos.

5.4 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 objetos dentro de la escena. Puede atender a varios objetos que haya ido detectando con el tiempo y almacenando en memoria, alternando entre ellos, aunque no estén dentro del campo de visión inmediato de la cámara. Los objetos pueden desaparecer eventualmente de la escena, con lo que deben ser eliminados del sistema para mantener la representación de la escena coherente con la realidad.

Para cumplir esta labor de olvido de antiguos elementos 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.

5.5 Implementación del sistema atentivo

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*). Y además, el mecanismo de arriba hacia abajo (*top-down*) existente es que los objetos relevantes son aquéllos que tienen una determinada apariencia: caras humanas, paralelogramos o flechas. Esta tendencia a mirar hacia objetos de cierto aspecto 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 diferentes pasos del algoritmo. Así, podemos distinguir cuatro estados:

0. Deliberar próximo objetivo (estado 0)
1. Completar movimiento sacádico (estado 1)
2. Analizar imagen (estado 2)
3. Seguir objeto (estado 3)

El funcionamiento es el siguiente. Inicialmente lo que hacemos, con el paso del tiempo, es ir actualizando los posibles objetos que tengamos ya almacenados en memoria. Por un lado comprobar si alguno de ellos ya está desfasado, porque su vida es inferior a un determinado umbral; y por otro, aumentar la saliencia y disminuir la vida.

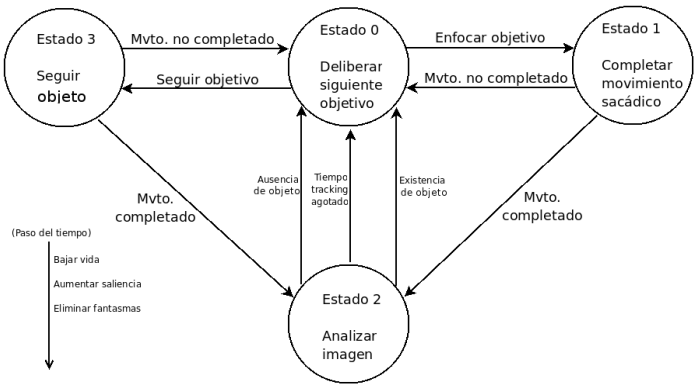

Figura 5.3: Diagrama de estados del sistema

Partiendo del estado inicial, el sistema se pregunta si hay algún objetivo a mirar (por si tenemos algún objeto previamente almacenado en memoria); en caso afirmativo, pasaremos al estado 1. En caso negativo, crearemos un nuevo punto de atención, y lo 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 estado 2 donde analizaremos si hay objetos relevantes 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 algún objeto, en cuyo caso se podrá hacer el seguimiento del mismo. Éste es precisamente el propósito de dicho estado.

6 Experimentos

Nuestros experimentos[1] han sido realizados con un robot real *Pioneer 2DX* de *ActivMedia Robotics* (ver *figura 6.2*), 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 (*Pantilt 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.

6.1 Reconstrucción del suelo

En este primer experimento el robot parte con un conocimiento nulo del entorno. Inicialmente, y como ya hemos comentado hará una exhaustiva exploración sistemática para obtener información del entorno. Así, el sistema tiene que comandar al cuello mecánico que realice movimientos sacádicos en busca de elementos por toda la escena. Estos movimientos son cortos, precisos y rápidos; lo justo para que de tiempo a examinar si hay o no *algo* de interés en la imagen actual recibida con la cámara. Transcurrido un cierto tiempo, el sistema comienza a detectar segmentos (*figura 6.1*).

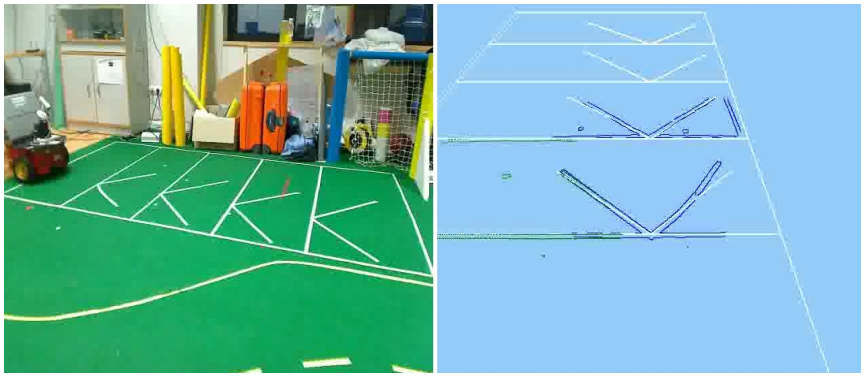

Figura 6.1: Fase inicial de la reconstrucción de líneas del suelo

1 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://jderobot.org/index.php/RobotVision

Tras varios vistazos el robot es capaz de reconstruir de forma plausible los segmentos detectados en su camino (*figura 6.2*). Continuamente la memoria se somete a un postprocesado gracias al cual obtenemos segmentos únicos, coincidiendo con lo que existe en la realidad (recordemos la *fig. 4.2*).

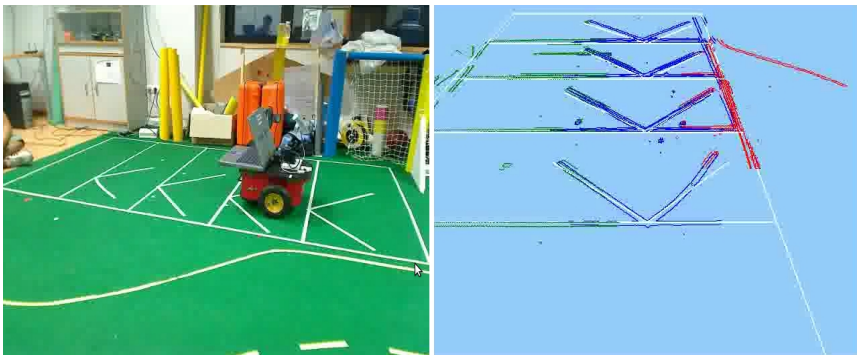

Figura 6.2: Fase final de la reconstrucción de líneas del suelo

6.2 Abstracción de paralelogramos

En el segundo experimento partimos igualmente del desconocimiento absoluto del entorno que rodea al robot. En este caso, además de encontrar segmentos por el entorno, el sistema puede abstraer paralelogramos dadas las características del conjunto de segmentos que encuentra en la escena.

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 el robot comienza a detectar objetos de interés en escena (ver *figura 6.3*).

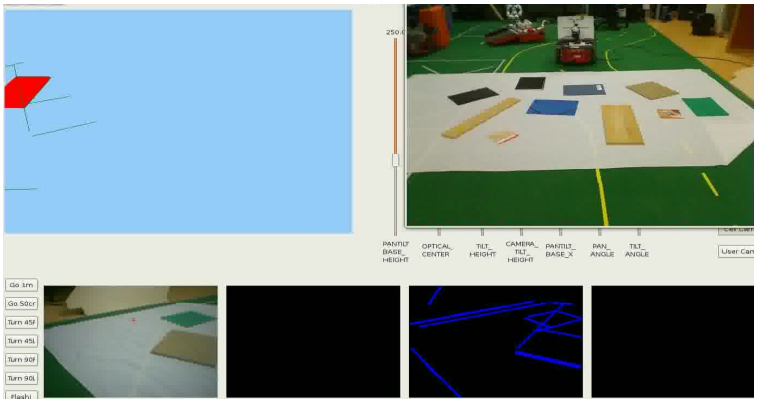

Figura 6.3: Fase inicial del reconocimiento de paralelogramos

Cuando comienza a haber varios elementos (paralelogramos) en memoria (ver *figura 6.4*), el tiempo de exploración forzada se ha aumentado. Este hecho nos permite mantener la mirada durante más tiempo a los objetos que tenemos detectados, así como seguir buscando otros posibles. Gracias a este mecanismo, podemos encontrar casi todos los elementos existentes en la escena (nótese que algunos objetos causan problemas de detección por su textura, lo que causa reflejos molestos al sistema y éste es incapaz de reconocerlos).

Lo que conseguimos con este incremento de tiempo es que a medida que vamos detectando más y más elementos, la búsqueda de otros nuevos se hará cada vez menos frecuente. No obstante, cuando toque hacer una exploración forzada, iremos igualmente a explorar nuevas zonas, volviendo después al elemento que se visitó hace más tiempo.

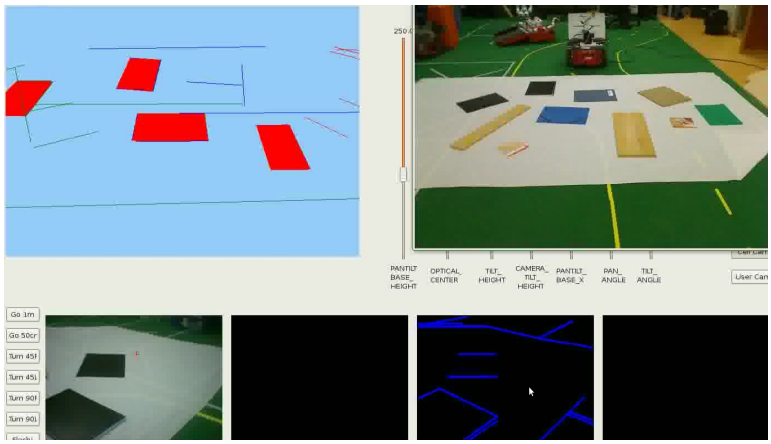

Figura 6.4: Fase final del reconocimiento de paralelogramos

6.3 Abstracción de flechas como marca de rumbo

En este último experimento nos basamos en las mismas ideas dadas en el anterior, pero en este caso nos centramos en el reconocimiento de flechas en el entorno, y su uso como marca de rumbo por pistas para la navegación del robot en un entorno sencillo, basada única y exclusivamente en visión. En la *figura 6.5* es el momento en el que el robot reconoce la flecha como tal, habiendo sido previamente detectados los segmentos que la conforman.

Dadas las características de la flecha, el sistema es capaz de abstraer el concepto flecha y representarla como tal en la memoria 3D (véase la flecha de color verde en la *figura 6.5*).

Una vez detectada, se marca automáticamente el rumbo del robot siguiendo la indicación de la misma (véase en la *figura 6.5* la línea de color amarillo que parte del robot).

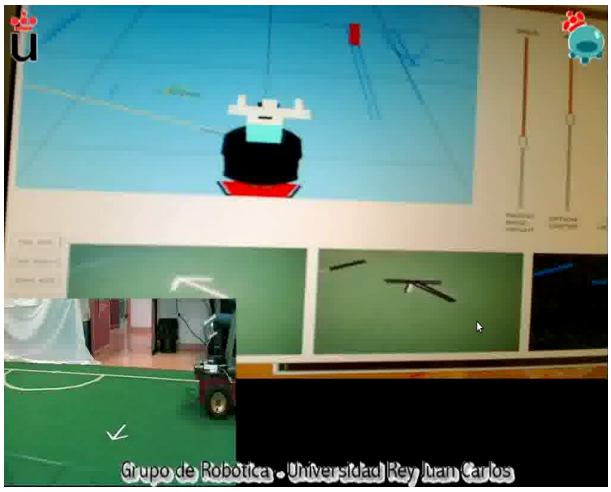

Figura 6.5: Reconocimiento de flechas como marca de rumbo

Finalmente, en la *figura 6.6* hemos mezclado objetos de distintas clases (paralelogramos y flechas) y aparecen reconocidos como tales en la memoria 3D visual. Asimismo, ante la detección de varias flechas en el entorno del robot, éste considera primordial la más cercana. De ahí que su rumbo ahora sea diferente, porque la nueva flecha está más cerca que la que tenía establecida anteriormente como marca de rumbo.

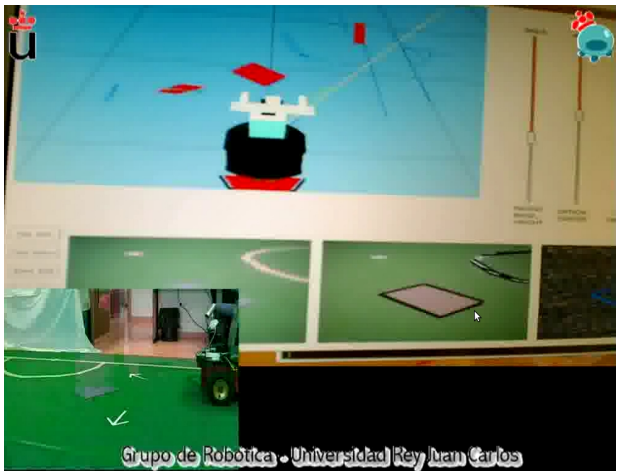

Figura 6.6: Reconocimiento de paralelogramos y flechas

7 Conclusiones

En este trabajo hemos presentado un sistema perceptivo visual cuyo propósito es encontrar objetos o conceptos abstractos por la escena circundante a éste, siguiéndolos con la mirada en tal caso. Para ello se ha desarrollado un mecanismo de dinámica concurrente entre vida y saliencia, en el cual el elemento en memoria con mayor saliencia es la siguiente en ser visitado y, por tanto, el que dirige el movimiento del *pan-tilt* en todo momento. Así hemos conseguido que el robot siga con la mirada a todos los objetos que hemos considerado de interés. 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 objetos 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 visual 3D de corto plazo. Esto ha facilitado la representación interna de la información que rodea al robot, ya que puede que los objetos estén situados en posiciones que el robot no es capaz de ver en un momento dado pero en las que *sabe* que existen elementos de interés.

Otro aspecto a destacar es el olvido de elementos 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 un objeto, 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.

Una de las posibles mejoras a este trabajo preliminar podría ser el uso de la atención visual para que el robot, aparte de reconocer y abstraer correctamente los objetos que tiene a su alrededor, navegue de forma autónoma apoyándose exclusivamente en la memoria visual, enriquecida con nuevos conceptos y/o primitivas.

El sistema podría mover la cámara para detectar los distintos elementos y marcas visuales de navegación (flechas), así como obstáculos potencialmente peligrosos (como pueden ser las paredes).

Una vez detectados todos los elementos, el sistema los incluiría en su representación interna del mundo, y repartiría la mirada entre todos ellos. Asimismo, y según la tarea a realizar, se podría modular el sistema atentivo para dedicar más o menos tiempo en procesar los distintos estímulos.

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., Martínez de la Casa, M., González, T. 2008. Overt visual attention inside JDE control architecture. International Journal of Intelligent Computing in Medical Sciences and Image Procesing. Volume 2, Number 2, pp 93-100, ISSN: 1931-308X. TSI Press, USA.

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.

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.

Nehmzow, U. 1993. Animal and robot navigation. The Biology and Technology of Intelligent Autonomous Agents.

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.

Vega, J., Cañas, J.M. 2009. Sistema de atención visual para la interacción persona-robot. In Workshop on Interacción persona-robot, Robocity 2030, pp. 91-110. ISBN: 978-84-692-5987-0.

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.