

ESTUDIO DE INTERSECCIONES SEMAFÓRICAS REGULADAS POR VISIÓN ARTIFICIAL

Cristina Conde (1), Jorge Pérez (1), Pedro González (1), Jesús Silva (1), Laura Vaquero (1), Licesio J. Rodríguez (1), Enrique Cabello (1), Jesús Monclús (2) y Tomás Santa Cecilia (2)

(1) Universidad Rey Juan Carlos
ESCET
C/ Tulipán S/N
28933 Móstoles (Spain)

(2) RACE
C/ Isaac Newton 4
28760 Tres Cantos (Madrid)

RESUMEN

En este proyecto se han desarrollado el diseño, la planificación y el montaje de un sistema para la regulación de intersecciones semafóricas mediante mecanismos de visión artificial. Para ello se ha diseñado y desarrollado un vídeo-sensor, orientado al análisis de imágenes de peatones y vehículos en un semáforo, y destinado al control inteligente de la iluminación del semáforo en función de diversos factores como: número de peatones presentes, situación de los mismos (en la acera o en pleno paso), número de vehículos, etc.

DESCRIPCIÓN INFORMÁTICA

La figura 1, muestra un esquema del proyecto. Se dispone de dos cámaras, una de ellas está dirigida hacia el paso de peatones y la otra hacia los coches que lleguen al mismo. Ambas están conectadas a un ordenador que continuamente está procesando la información que le llega y en función de la cual manda la señal oportuna al autómata, que será el que controle al propio semáforo.

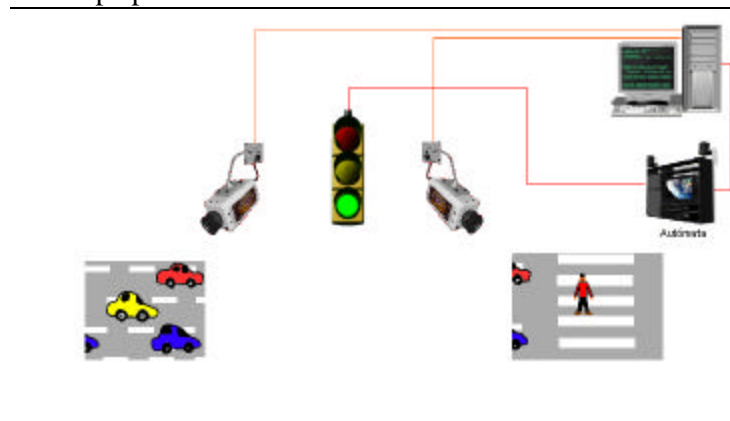


Figura 1. Esquema del sistema.

Estas señales van a depender de la información detectada a través de las cámaras. Se pretende controlar la presencia de peatones o de coches en la vía pública

para evitar por ejemplo la existencia de peatones esperando para cruzar el paso de cebrá sin que haya coches que se aproximen hacia el mismo.

Para llegar al punto en el que a partir de dos imágenes, obtenidas una por cada cámara, se envíe una señal al autómata, es necesario procesar ambas imágenes para obtener información sobre los peatones o los coches. Dependiendo de la cámara con la que se haya capturado y en función de la presencia de unos y otros se enviará la señal oportuna al autómata. Para ello se ha utilizado el algoritmo de alto nivel mostrado en la siguiente figura.

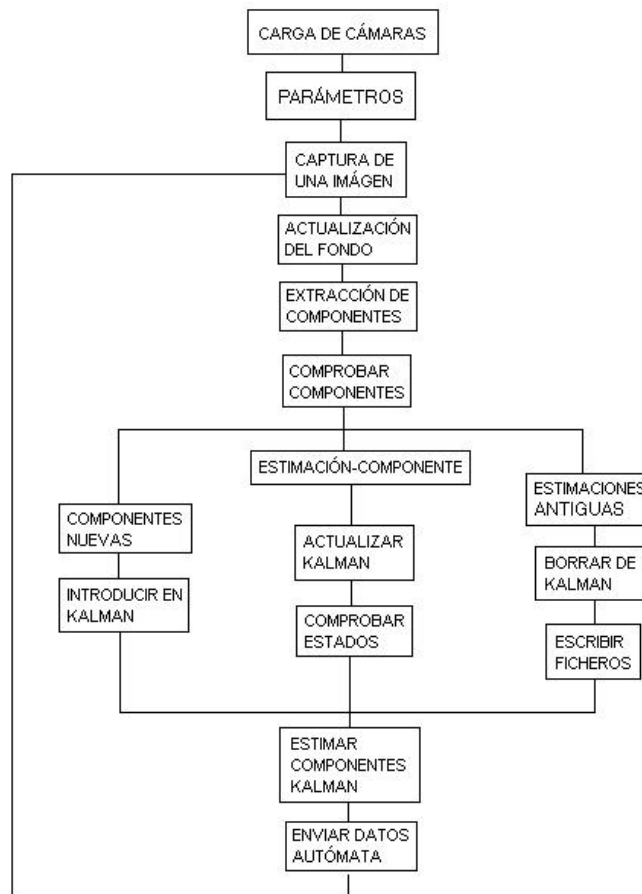


Figura 2. Esquema en alto nivel del Algoritmo utilizado

El primer paso que se tiene que llevar a cabo antes de empezar a capturar y procesar imágenes, es efectuar la carga de los parámetros de configuración y de las cámaras. La posibilidad de variar los parámetros es indispensable ya que permite que la aplicación sea totalmente portable y se pueda adaptar a cualquier tipo de intersección semafórica.

Una vez realizada la carga de los parámetros ya se puede empezar a capturar imágenes en tiempo real. Se va a adquirir una imagen de cada cámara, los objetos de una serán peatones mientras que los de la otra serán considerados vehículos. Por este motivo se van a mantener dos procesos con dos sucesiones distintas de imágenes a las que se les van a aplicar los mismos algoritmos. Ambos procesos se diferenciarán en los tipos de objetos extraídos, en un caso peatones y en otro coches.

Para poder realizar la obtención de objetos que se muevan en una imagen resulta necesario diferenciar entre los objetos que se mueven y los que forman el fondo de la imagen. Por esta razón el primer problema que se plantea es la obtención de las componentes sobre un fondo uniforme (aquel que tiene todos sus píxeles vacíos). Esto se realiza hallando la imagen de fondo que contiene todo aquello que ha permanecido invariable a lo largo del tiempo, en este caso a lo largo de varias imágenes. Téngase en cuenta que al tratarse de una situación totalmente dinámica, este fondo irá variando continuamente por lo que se debe ir actualizando con cada imagen recibida.

Por esta razón, la solución al problema planteado es la realización de operaciones con píxeles comunes a varias imágenes para así obtener una única imagen que se corresponda con el fondo. Dos de las formas de operar con los píxeles, ambas implementadas y con posibilidad de ser escogidas por el usuario son: la media, de complejidad $O(n)$, y la moda, de complejidad $O(n^2)$. De píxeles comunes de varias imágenes se obtiene la media o la moda respectivamente. El número de imágenes y el método empleado se deja como opción al usuario final, y una vez elegido permanecerá constante a lo largo de la ejecución de la aplicación. Se recomienda utilizar la moda siempre que el número de imágenes empleadas para el cálculo del fondo sea relativamente pequeño.

Antes de aplicar al conjunto de imágenes la media o la moda, es necesaria la actualización de éstas. Dado que el número de imágenes que se necesitan para la extracción del fondo debe permanecer siempre constante se debe utilizar un algoritmo de reemplazo. El algoritmo empleado es una pila LIFO. De esta manera tendremos siempre un fondo actualizado en cada momento.



Fig.3 Imagen Original Resultado

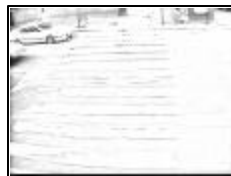


Fig. 4. Imagen Fondo



Fig. 5. Imagen Resultado

Una vez obtenida la imagen fondo, para poder obtener aquellos objetos que se han movido en la imagen que fue capturada bastará con realizar la resta píxel a píxel. De esta manera se obtiene una nueva imagen (Fig. 5) con los objetos buscados sobre un fondo blanco.

Una vez calculada la imagen que representa a aquellos objetos que se han movido, sobre un fondo blanco, el siguiente paso es extraer esos objetos a los que se pasará a llamar componentes. Para ello se utilizará un paso previo que extraerá toda componente que esté completamente rodeada por píxeles de color blanco que serán consideraremos como “vacíos”.

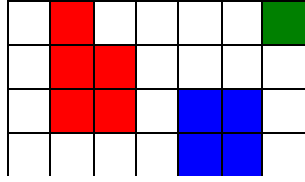


Fig 6. Imagen con componentes.
componentes=3.

	1				3
	1	1			
	1	1	2	2	
			2	2	

Fig 7. Imagen etiquetada. N° de

Estas componentes tienen como característica fundamental la interconexión de todos sus píxeles. Se utiliza un método en el que se recorre la imagen pixel a pixel comprobando sus vecinos y etiquetando de forma tal que cada etiqueta represente a una única componente. De esta manera obtendremos una nueva imagen “etiquetada” que va a indicar donde están los objetos que se buscan, es decir aquellos que se movieron.

Una vez obtenidas las componentes que se han movido en la nueva imagen es necesario saber cual ha sido su trayectoria, por ejemplo para ignorar vehículos que se alejen del paso de peatones. El calcular su trayectoria conlleva realizar una asociación entre los objetos obtenidos en este paso y los que fueron obtenidos en anteriores imágenes capturadas. Para esto se utiliza el filtro de Kalman.

El filtro de Kalman se basa en el análisis de las diferencias entre las observaciones registradas asociadas a una componente dada y las posiciones que el modelo numérico prevé. El objetivo es hacer pronóstico de esta discrepancia y así corregir el valor previsto por el modelo numérico para que se ajuste más a las características de la componente. Mediante este mecanismo, es el modelo el que se va adaptando a la componente, corrigiéndose según el error cometido e intentando en cada paso ser más preciso.

El filtro de Kalman consta de dos pasos: predicción y corrección. Estos pasos se van a aplicar a la detección de componentes en la imagen. Al aplicar Kalman se buscan componentes en zonas determinadas y no en toda la imagen. Beneficiándose así el rendimiento del sistema.

Como se ha dicho, el principal objetivo de Kalman aplicado al caso que se trata, es establecer una proximidad entre una componente y su estimación (predicción), de esta manera se podrá realizar fácilmente una correspondencia entre todas las componentes que se tengan en la pantalla y las estimaciones generadas en el paso anterior. Lográndose así realizar de una forma fiable el seguimiento de las mismas a lo largo de varias imágenes.

En nuestro caso el funcionamiento de Kalman consistirá en, una vez diferenciadas las componentes de una imagen introducirlas en el filtro. Éste, se encargará de realizar la estimación para el siguiente paso. Una vez en la siguiente imagen se pasará a la fase de actualización, es decir que una vez recuperados los datos reales de las componentes estimadas se introducirán en el filtro y éste se irá adaptando teniendo en cuenta el error cometido. Una vez actualizado se pasará a realizar una nueva estimación, continuando el ciclo hasta que la componente estimada desaparezca de la imagen, quedando así concluido el filtro para esa componente en particular.

Conocidas tanto las estimaciones como las componentes se deberá realizar la correspondencia entre ellas para poder pasar los nuevos datos al filtro de Kalman. Además no sólo se tendrá que asignar a cada estimación una componente sino que deberá ser capaz de descubrir componentes nuevas, aquellas para las que no hay estimación y estimaciones para las que no hay componente. En ese caso se dirá que la componente ha desaparecido de la imagen y se tomarán las acciones que se consideren oportunas como puede ser escribir su recorrido (trayectoria, velocidades y frames) en un fichero, etc.

El método únicamente se basa en la distancia euclídea a las componentes. Halladas todas las posibles combinaciones de estimaciones-componentes, se ordenan por distancias a componentes dentro de cada estimación. La primera entrada querrá decir que para esa estimación es la componente con menor distancia y que por lo tanto es la que más posibilidades tiene de ser la buscada. Las siguientes imágenes muestran su resultado.

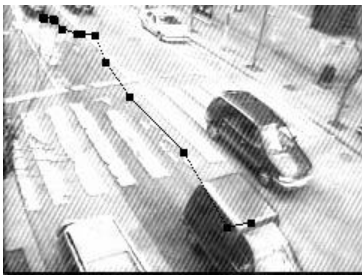


Fig 8. Trayectoria de un coche

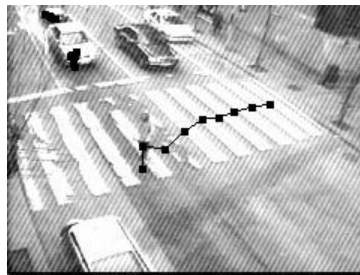


Fig 9 y Fig 10. Trayectoria de un peatón

Como se comentó, el propósito del programa es su aplicación a una intersección semafórica, por lo que es necesario controlar los posibles estados de cada cámara y agruparlos en un único objeto que será el que mande la señal oportuna al autómata en función del modo en el que se encuentre.

Así, cada vez que se actualicen las componentes introducidas en Kalman, se comprobará la vida de éstas, es decir el número de imágenes en las que ha estado presente y si superan un mínimo. Entonces, se pasará a comprobar de qué tipo de objeto se trata en función de su posición, de la trayectoria llevada y de la cámara en la que fue detectado.

Por este motivo es necesario delimitar zonas, y en función de la zona en la que esté presente será considerado como un peatón o como un coche.



Fig 11. Original



Fig 13. Original

Fig 12. Con Zonas



Fig 14. Con Zonas

Estas zonas están delimitadas por puntos que pueden variarse de forma sencilla desde la interfaz de la aplicación. Además como se ha dicho a partir de ellas se podrá saber de qué clase de objeto se trata cada componente pudiéndose definir los modos de funcionamiento establecidos en los objetivos y que se muestran a continuación.

Camara 0	Camara 1		Modo
	<i>Peatones Esperando</i>	<i>Peatones Cruzando</i>	
<i>Coches</i>			
SI	NO	NO	3
SI	SI	NO	1
NO	SI	NO	2
SI	SI	SI	1
NO	SI	SI	2
SI	NO	SI	2
NO	NO	SI	2
NO	NO	NO	1

Fig 15. Modos de funcionamiento para el Automata.

La siguiente figura marca el efecto de los modos mandados al autómata sobre el propio semáforo. El “Modo 1” es una transición de tres períodos, mientras que el resto es de un único período.

Modo 1	Vehículos	Verde	Amarillo	Rojo
	Peatones	Rojo	Rojo	Verde
Modo 2	Vehículos	Verde		
Modo 3	Peatones	Rojo		
	Vehículos	Rojo		
	Peatones	Verde		

Fig 16. Efecto de los modos del autómata en los colores del Semáforo

PRUEBAS, RESULTADOS Y CONCLUSIONES

Durante el desarrollo de la aplicación se fueron realizando un conjunto de experimentos para ir probando los nuevos algoritmos, así como una prueba completa al concluirlo. En un principio las pruebas iniciales fueron realizadas en condiciones de laboratorio, usando vehículos de radio control pudiendo así simular la presencia de peatones y/o coches. Por lo tanto esta prueba inicial fue realizada en ambientes controlados, donde los cambios de luz no eran abruptos, el movimiento era suave y el número de objetos no era muy grande.

Una vez que los resultados fueron satisfactorios en condiciones de laboratorio, se realizaron un segundo conjunto de pruebas en el circuito del Jarama, en una pista propiedad del Real Automóvil Club de España (RACE). Estas pruebas fueron llevadas a cabo durante los meses de Mayo, Junio y Julio de 2003. Durante las mismas, las cámaras fueron colocadas ante un tráfico ligero, alrededor de 20 niños usaron las instalaciones (ver las Figuras 17 y 18). La mitad, usaban bicicletas simulando la presencia de coches y los demás actuaban como peatones. Así mismo además de probar el funcionamiento del sistema en tiempo real se capturaron videos para su posterior análisis en el laboratorio.

Todas las pruebas han sido realizadas sólo durante el día y en condiciones de visibilidad excelentes. Durante el otoño de 2003 se seguirán realizando nuevas pruebas teniendo en cuenta el cambio de condiciones meteorológicas (la lluvia, la niebla etcétera).



Figura 17: Peatones cruzando la calle
cámara



Figura 18: Ciclistas vistos por la
cámara

Una vez que esta fase del proyecto haya terminado, nuestro trabajo estará enfocado en probar el sistema en condiciones meteorológicas extremas, e intentar integrar más semáforos en el sistema, pudiendo ser completado con otra clase de sensores como pulsadores, etc.

Este proyecto ha sido una iniciativa conjunta entre la Universidad Rey Juan Carlos y el Real Automóvil Club de España, dirigida a pruebas de algoritmos de visión artificial analizando sus capacidades en situaciones reales, y su aplicabilidad y utilidad en condiciones de tráfico. La visión de ordenador es en nuestros días una tecnología avanzada válida para implementarla y usarla en calles reales para controlar condiciones de tráfico con información en tiempo real. Así mismo, estas tecnologías contribuyen a la mejora de la seguridad de tráfico y a ciudades más seguras, y nuestra obligación es

investigar en esta dirección: el desarrollo de la tecnología que hace ciudades más confortables y ciudadanos más seguros.

BIBLIOGRAFÍA

- [1] European Transport Safety Council (ETSC) <http://www.etsc.be/stats3.ppt>
- [2] "Encuesta RACE sobre Movilidad en Grandes Ciudades: La Ciudad Más Habitable". Press Report. Departamento de Seguridad Vial, Real Automóvil Club de España. Madrid (Spain), 2000.
- [3] European Transport Safety Council (ETSC) <http://www.etsc.be/stats1.ppt>
- [4] International Road Traffic and Accident Database (IRTAD) <http://www.bast.de/irtad>
- [5] "Anuario Estadístico Accidentes 2000 y 2001". Dirección General del Tráfico, Ministerio del Interior. Madrid, 2001-2002.
- [6] "Infracciones de tráfico de conductores y peatones". Revista Consumer. <http://www.revista.consumer.es>
- [7] R. Cutler and L. Davis. *Real-Time Periodic Motion Detection, Analysis and Applications*. Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol 2, Pp 326-331. 1999.
- [8] C. Curio et al. *Walking Pedestrian Recognition*. IEEE Trans. Intelligent Transportation Systems. Vol 1, N 3, Pp 155-163. 2000.
- [9] S. Kang, H. Byun, S. Lee. *Real-Time Pedestrian Detection Using Support Vector Machines*. International Journal of Pattern Recognition and Artificial Intelligence. Vol 17, N 3, Pp 405-416. 2003.
- [10] P. Carrea and G. Sala. *Short Range Area Monitoring for Pre-crash and Pedestrian Protection: The Chameleon and Protection Projects*. Proc 9th Aachener Colloquium Automobile and Engine Technology. Pp. 629-639. 2000.
- [11] D. M. Gavrilu. *Sensor-Based Pedestrian Protection*. IEEE Intelligent Systems, Vol 16, No 6. Nov-Dec 2001.
- [12] N. Obolensky, D. Erdogmus, J.C. Principe. *A Time-Varying Kalman Filter Applied to Moving Target Tracking*, Proceedings of CONTROL'02, pp. 418-422. 2002
- [13] N. Ayache. *Vision Stéréoscopique Et Perception Multisensorielle, Applications à la robotique mobile*, InterEditions, pp. 176-192, 264-266. 1989