

visrob

Gonzalo Gismero

Agenda

- Introducción a Visión por Computador
- Proceso de desarrollo
- Implementación
 - Arquitectura de la solución
 - Agrupación de blobs
 - Reconocimiento de objetos
 - Extracción de distancia y ángulo
 - Niveles de confianza
 - Filtro de Zigbee
 - Cooperación
- Pruebas realizadas

Agenda

- Ventajas y desventajas
- Trabajo a futuro

Introducción a Visión por Computador

Proceso de extracción de información de una imagen, para resolver una tarea específica.

Proceso de visión:

1. Segmentación de imagen.
2. Identificación de objetos.
3. Extracción de propiedades para los objetos identificados.

Proceso de desarrollo

Iterativo incremental, con las siguientes etapas:

1. Investigación de algoritmos
2. Desarrollo de capas
3. Prueba unitaria de capas
4. Integración
5. Realizar documentación

Fuera del proceso iterativo:

6. Pruebas del sistema

Proceso de desarrollo

Dos iteraciones del proceso:

- **Visión cartesiana:**

Se retornan los datos de los objetos de la escena en coordenadas cartesianas, localizando al agente.

- **Visión polar:**

Se retornan los datos de los objetos de la escena en coordenadas polares, sin localizar al agente.

Proceso de desarrollo

Tareas realizadas:

1. Investigación y pruebas de algoritmos de visión, reconstrucción de escenas 3D, robótica cooperativa y algoritmos de localización. (4 meses)
2. Desarrollo de capa de Interfaz HaViMo, desarrollo de simulador de HaViMo. (3 meses)

Proceso de desarrollo

3. Desarrollo de primera versión de capa de Visión (cartesiana), y capa de Cooperación. Investigación de Zigbee. (3 meses)
4. Investigación de algoritmos de visión polar y pattern matching. (2 meses)
5. Desarrollo de visRobUtils, investigación de Zig2Serial, pruebas unitarias de cooperación. (2 meses)
6. Documentación (Estado del arte, SUMO, Informe Final). (3 meses)

Proceso de desarrollo

7. Desarrollo de segunda versión de capa de Visión (polar). (3 meses)
8. Desarrollo de segunda versión de capa de Cooperación e integración con capas inferiores. (1 mes)
9. Pruebas del sistema y correcciones. (1 mes)

Implementación

Modelo de capas:

- Interfaz con HaViMo
- Visión
- Visión Cooperativa

Posibilita distribución de trabajo, facilita testing unitario, permite reutilización de capas para otros fines.

Arq - Capa de Interfaz con HaViMo

Compuesta por un único módulo que implementa funciones básicas sobre el dispositivo de hardware:

- Orden de procesamiento del *frame*
- Copia de datos procesados desde memoria del dispositivo a memoria del controlador.

HaViMo realiza por hardware la etapa de segmentación del proceso de visión.

Arq - Capa de Visión

Compuesta por 3 módulos que realizan las etapas del proceso de visión restantes:

- *Clasificación:*
Agrupación y filtrado de *blobs*
- *Object Recognition:*
Identificación de objetos y extracción de propiedades.
- *Vision:*
Encapsulamiento de operaciones

Arq - Capa de Visión Cooperativa

- *Vision Communication:*

Transforma y transmite mensajes en su codificación de enteros, por el filtro de zigbee y viceversa.

- *Object Buffer:*

Almacena temporalmente los datos recibidos.

- *Cooperative Vision:*

Invoca la visión local, enviando los objetos identificados y procesando mensajes recibidos

Arq - Módulos extra

- *dynamixel_custom*:

Basado en la librería *libdynamixel* provista por BIOLOID para poder comunicar el controlador CM-510 con el dispositivo HaViMo.

Arq - Módulos extra

- *zigbeeVisionFilter*:

Módulo implementado sobre la librería *libzigbee* de BIOLOID. Permite diferenciar mensajes de la estrategia y de la visión, añadiendo a los mensajes de visión un *payload* de autenticación.

Arq - Módulos extra

- *zgb_console*:

Permite imprimir en una consola externa utilizando Zig2Serial. Desarrollado para *debugging*, a raíz del conflicto entre serial y *zigbee*.

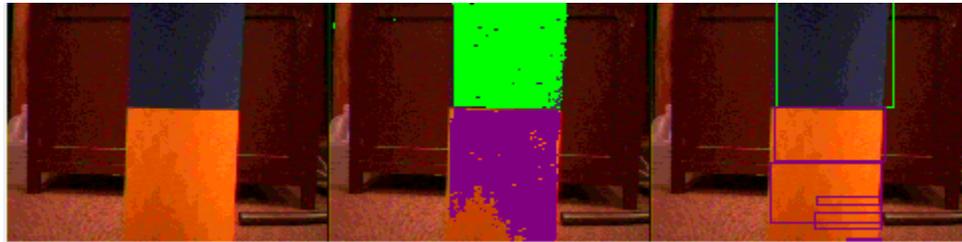
Agrupación de blobs

La agrupación de *blobs* es necesaria para contemplar casos de mala calibración y variación de condiciones de luz.

Se consigue eliminar *blobs* de ruido, reforzando los *blobs* de mayor tamaño detectados, según un umbral ajustable.

Agrupación de blobs

Segmentación realizada por hardware de un *landmark*, según la calibración que muestra la imagen central:



Reconocimiento de objetos

Tarea central de la capa de visión.

Abarca:

- Identificación de objetos
- Distancia al objeto
- Ángulo del frente del robot al objeto

Utiliza el concepto de "sistema de confianza", implementado en la práctica por equipos.

Reconocimiento de objetos

Bola:

Como parámetros para la identificación de la bola fueron tomados el tamaño del *blob*, así como cuán cuadrado es.

Se completan *blobs* sobre el margen inferior.

Reconocimiento de objetos

Landmarks:

Se tomó como parámetros la distancia entre los distintos *blobs* de colores que lo componen.

Se realizán aproximaciones de la altura que debería tener el *landmark* según la altura de cada uno de los *blobs*.

Reconocimiento de objetos

Arcos:

Identificados a partir de la posición de los 3 postes que lo componen.

Se realizan controles sobre la distancia del travesaño a los postes, filtrando así falsos positivos.

Reconocimiento de objetos

Postes:

Retornados únicamente cuando no se identifica el Arco de su equipo.

Agrupar menos rigurosamente *blobs* alineados verticalmente.

Reconocimiento de objetos

Robots:

La identificación de robots resulta difícil en la práctica dada la amplia variedad posible de robots.

Se utilizó un modelo probabilístico, bajo el concepto de qué probabilidad se tiene de reconocer un conjunto de *blobs* como un robot, dado que el objeto en la imagen es un robot.

Reconocimiento de objetos

Como entrada previa se realizan controles sobre la disposición de los *blobs* en la imagen, de las regiones pertenecientes a un robot.

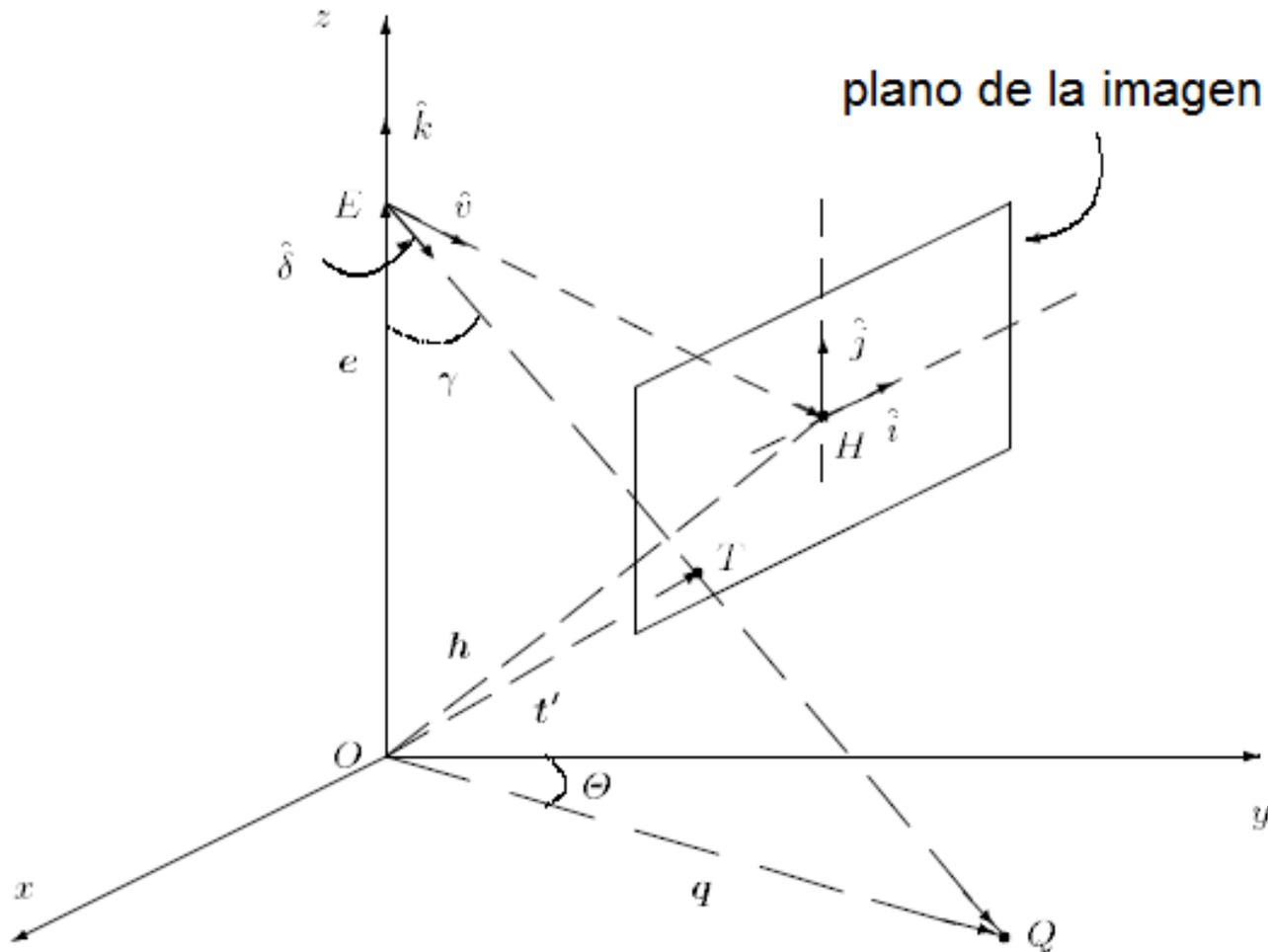
De esta manera se obtuvo un algoritmo de identificación, el cual se acopla de manera natural al sistema de confianza propuesto y no descarta posibles robots, por muy baja que sea su probabilidad.

Extracción de distancia y ángulo

Se utiliza un modelo vectorial, bajo la premisa que los objetos del campos se encuentran sobre el suelo.

De esta manera el problema se reduce a encontrar el punto proyectado del plano de la imagen sobre el plano del piso, con origen la cámara.

Extracción de distancia y ángulo



Extracción de distancia y ángulo

- El plano xy es el campo de juego.
- O es la proyección de la cámara perpendicular al plano del suelo.
- El vector e refiere a la altura de la cámara.
- El plano ij es el plano de la imagen capturada
- Q es el punto del objeto de interés para el que se obtendrá la distancia (q) y el ángulo al robot (θ)
- T es el punto Q en el plano de la imagen

Extracción de distancia y ángulo

Es necesario conocer la altura del robot, así como la inclinación de la cámara tanto horizontal, como verticalmente.

El resultado obtenido es robusto en cuanto al ángulo, pero no en cuanto a la distancia, siendo el mayor factor de error los errores en la inclinación vertical de la cámara.

Niveles de confianza

Debido al uso de un sistema de confianza, es necesario que cada objeto identificado cuente con un valor que describa la veracidad de la identificación.

Este valor (0-15) es utilizado además por la capa de Cooperación, informando de objetos que superen cierto nivel.

Filtro de Zigbee

Desarrollado como capa externa de zigbee, para poder obtener separadamente mensajes de visión y mensajes de estrategia, de manera transparente.

El mismo utiliza la librería de BIOLOID **libzigbee** para realizar la comunicación entre dispositivos.

Cooperación

Enfoque *result sharing*:

Comparte información de visión local a los demás aliados.

Se envía un mensaje RESET, limpiando los mensajes anteriores, seguido de mensajes BROADCAST con información de la visión local.

Cooperación

Para la transmisión de mensajes fue necesaria su codificación en cadenas de 16 bits, esta restricción es inherente a la tecnología zigbee.

Mensaje RESET:

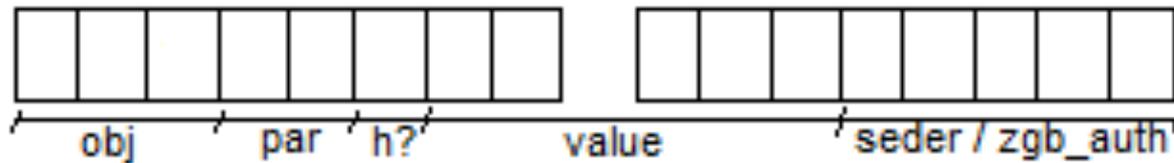
0	0	0	0	0	1	1	1
---	---	---	---	---	---	---	---

1	1	1					
---	---	---	--	--	--	--	--

seder / zgb_auth

Cooperación

Mensaje BROADCAST:



Donde:

- obj es el tipo de objeto (aliado, rival, bola, arco aliado, arco rival, poste aliado, poste rival)
- par es el tipo de parámetro (x, y, z)

Cooperación

- h? especifica si se trata de la parte alta (1) o baja del valor (0)
- value es el valor del parámetro
- sender es el agente que envía el mensaje
- zgb_auth es el *payload* del filtro de zigbee

Debido al tamaño de los mensajes, cada valor necesita de 2 mensajes para ser transmitido.

Cooperación

Son enviados como máximo 25 mensajes por *broadcast* realizado.

El *broadcast* no se efectúa en cada *frame* procesado.

Los mensajes poseen un tiempo de vida en los demás agentes.

Pruebas realizadas

1) Reconocimiento de objetos estáticos, desde una posición estática.

Primer escenario (75 iteraciones):

Objeto	Distancia	Ángulo	% detección	Distancia detectada	Ángulo detectado	Confianza
Rival	126 cm	+ 7.4	73 %	122 cm	+ 6.9	2.2
Bola	89 cm	- 18.3	82.5 %	86.2 cm	- 19.3	14.9

Pruebas realizadas

Segundo escenario (75 iteraciones):

Objeto	Distancia	Ángulo	% detección	Distancia detectada	Ángulo detectado	Confianza
Landmark	134.5 cm	+ 11	53 %	130.9 cm	+ 11	15

Tercer escenario (100 iteraciones):

Objeto	Distancia	Ángulo	% detección	Distancia detectada	Ángulo detectado	Confianza
Arco	224 cm	+ 5	39 %	250 cm	+ 1	13

Pruebas realizadas

Cuarto escenario (50 iteraciones):

Objeto	Distancia	Ángulo	% detección	Distancia detectada	Ángulo detectado	Confianza
Poste	92 cm	0	88 %	94 cm	- 0.5	4

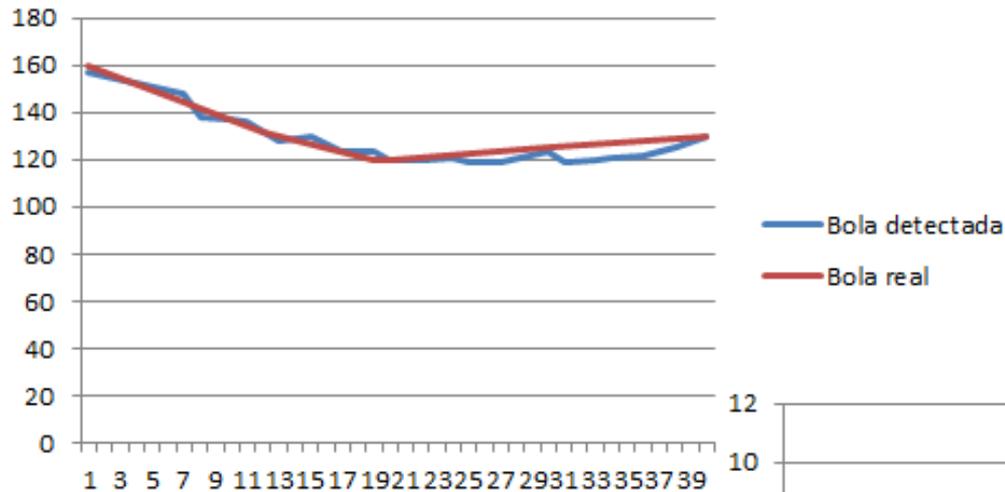
Pruebas realizadas

2) Reconocimiento de un robot y bola en movimiento, agente estático.

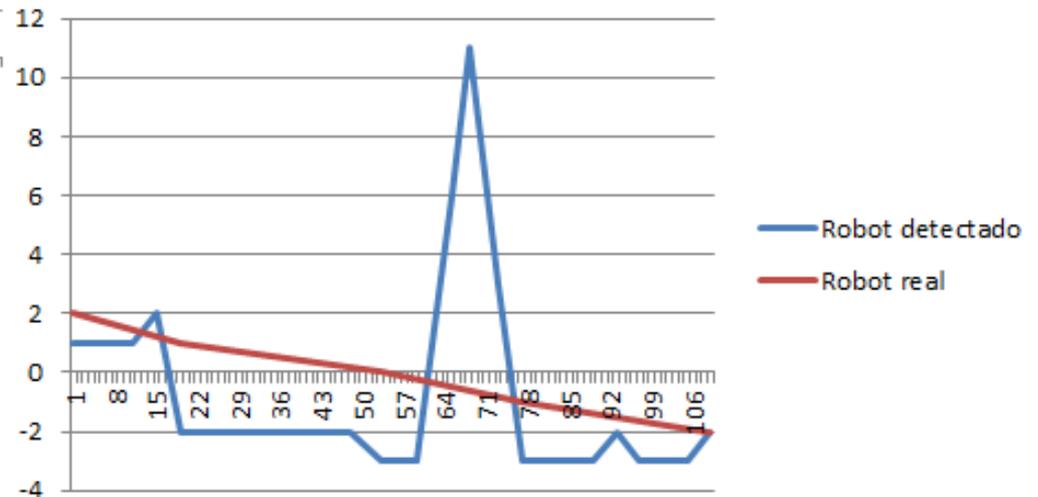


Pruebas realizadas

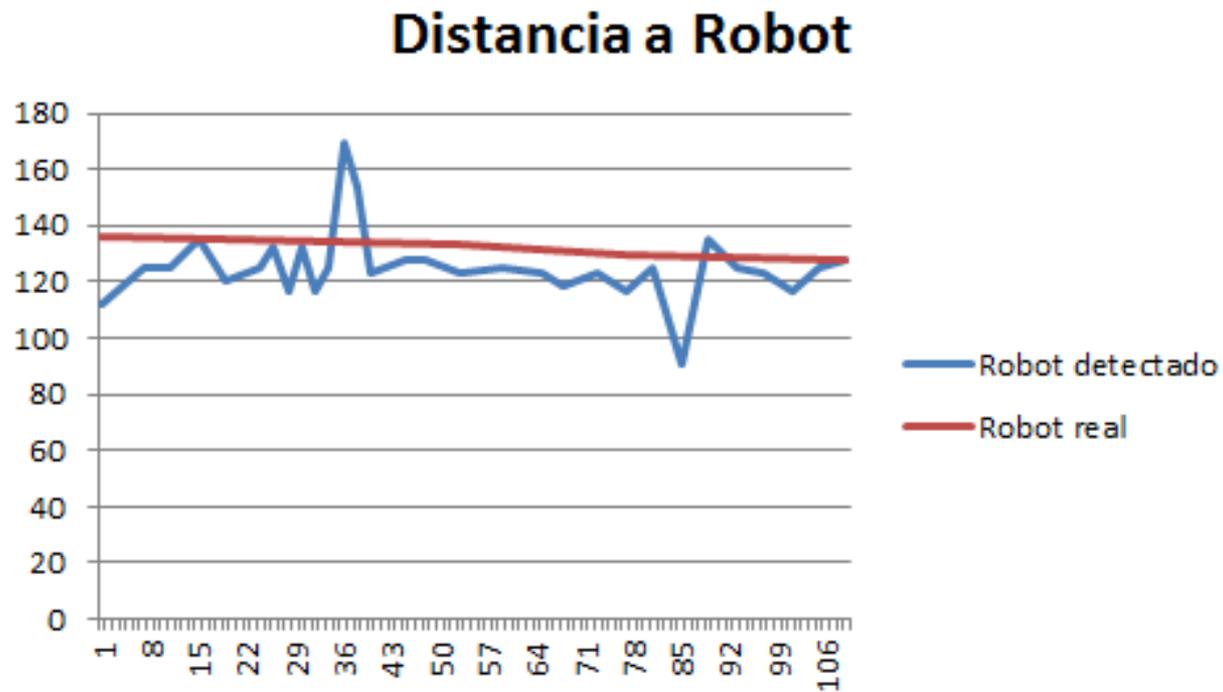
Distancia a Bola



Ángulo a Robot



Pruebas realizadas



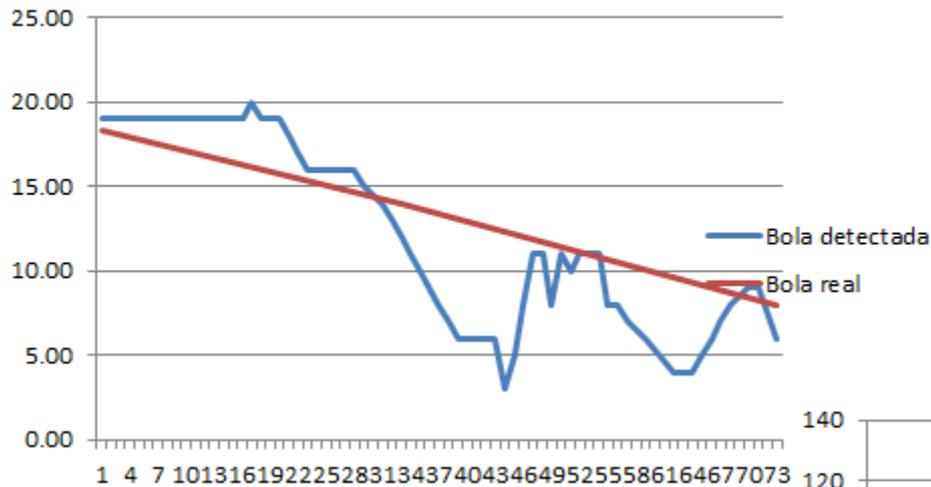
Pruebas realizadas

3) Reconocimiento de objetos estáticos, mientras el agente se encuentra en movimiento.

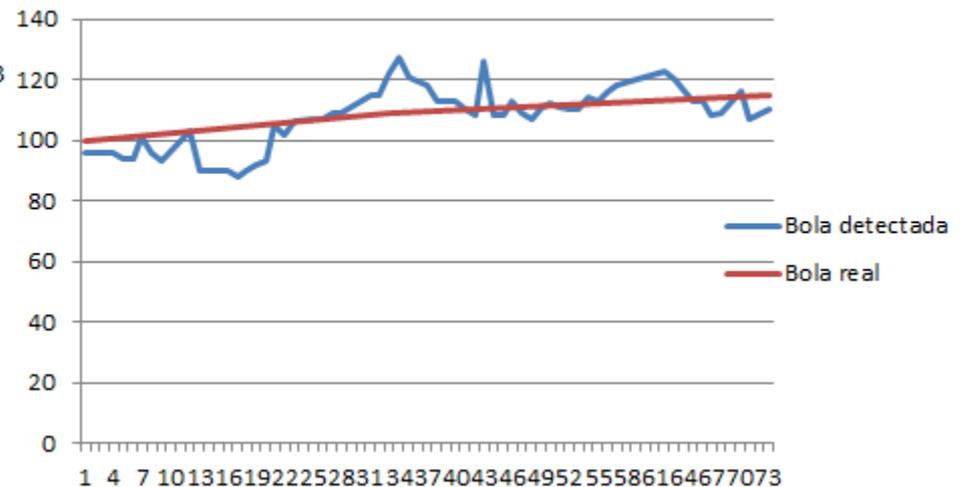
Realizado para todos los objetos de la cancha.

Pruebas realizadas

Ángulo a bola

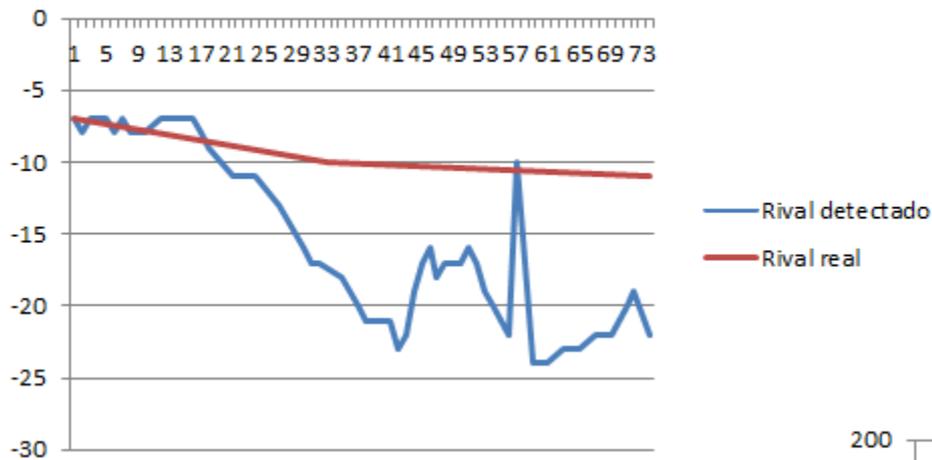


Distancia a bola

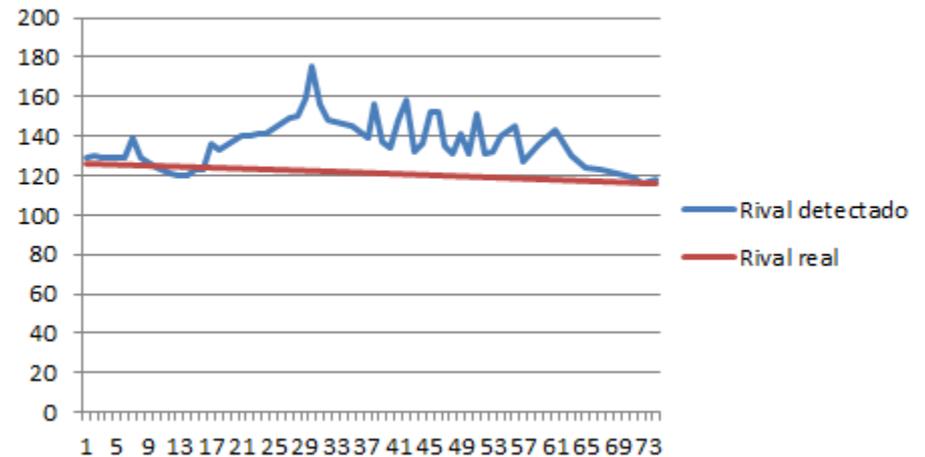


Pruebas realizadas

Ángulo al Rival

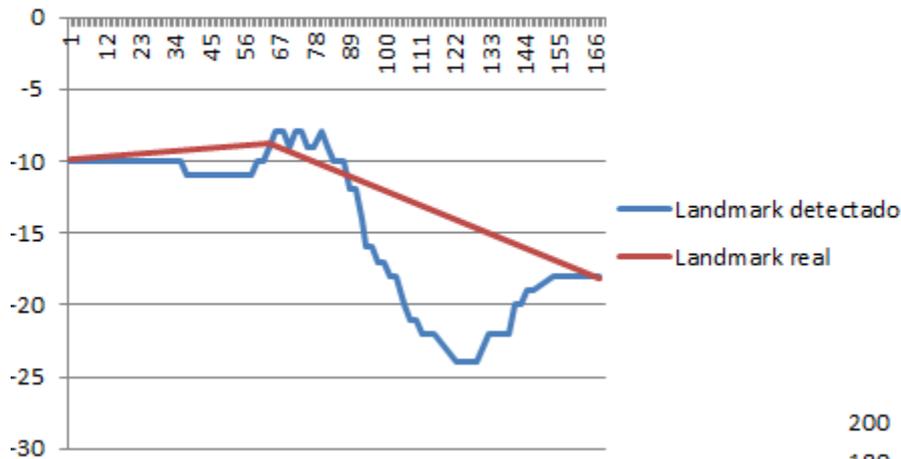


Distancia al Rival

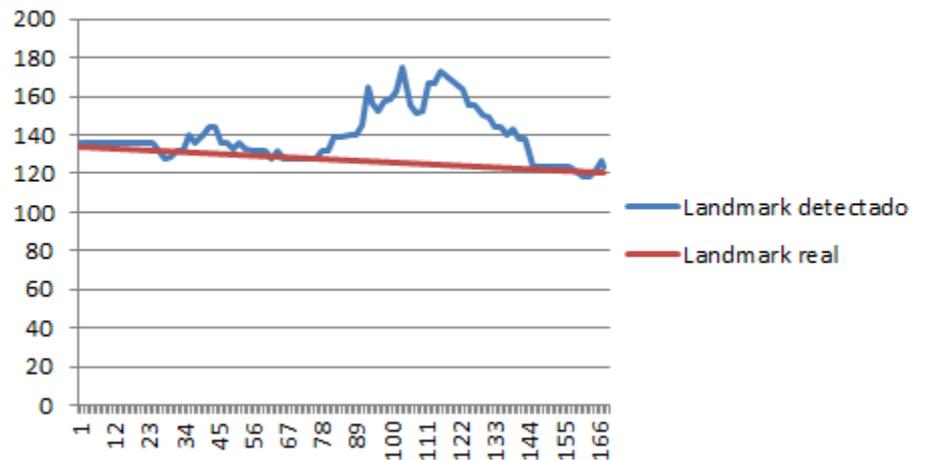


Pruebas realizadas

Ángulo al Landmark

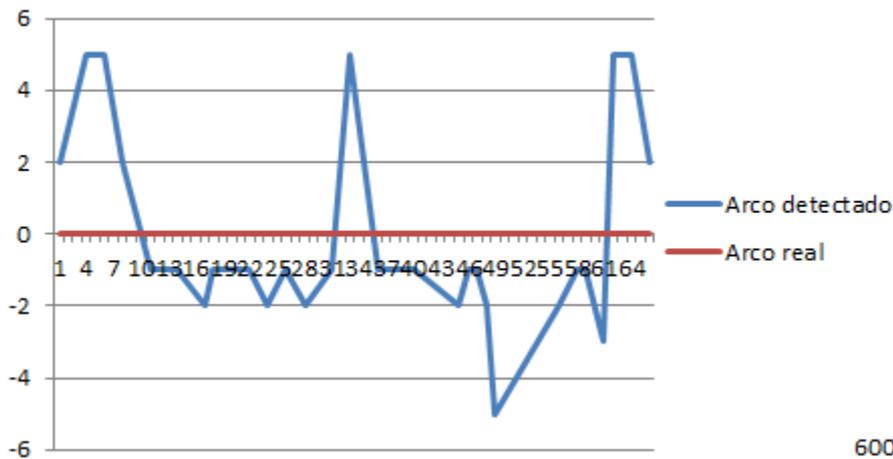


Distancia al Landmark

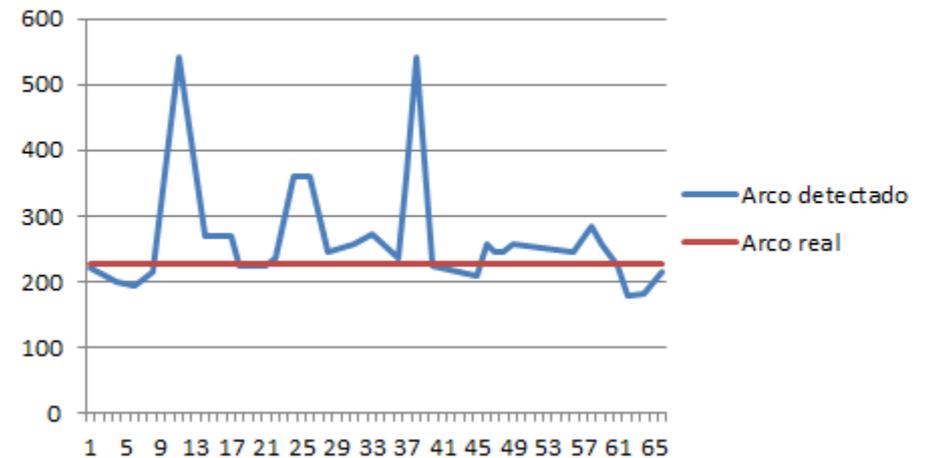


Pruebas realizadas

Ángulo al Arco

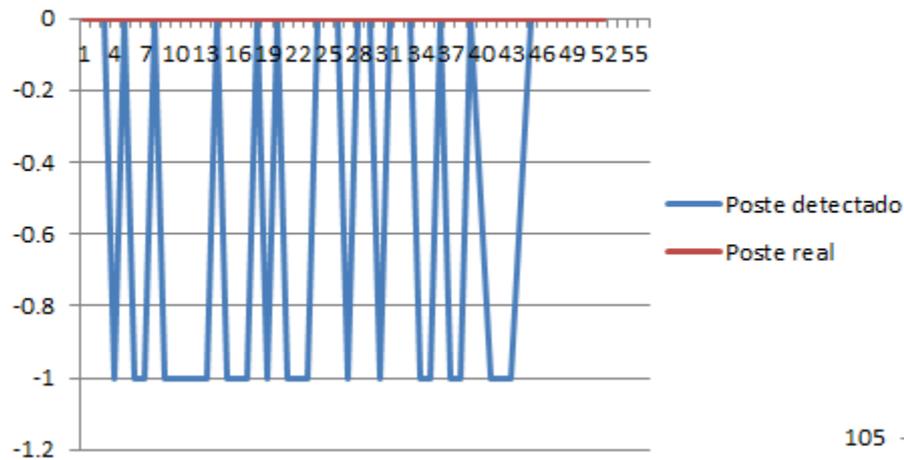


Distancia al arco

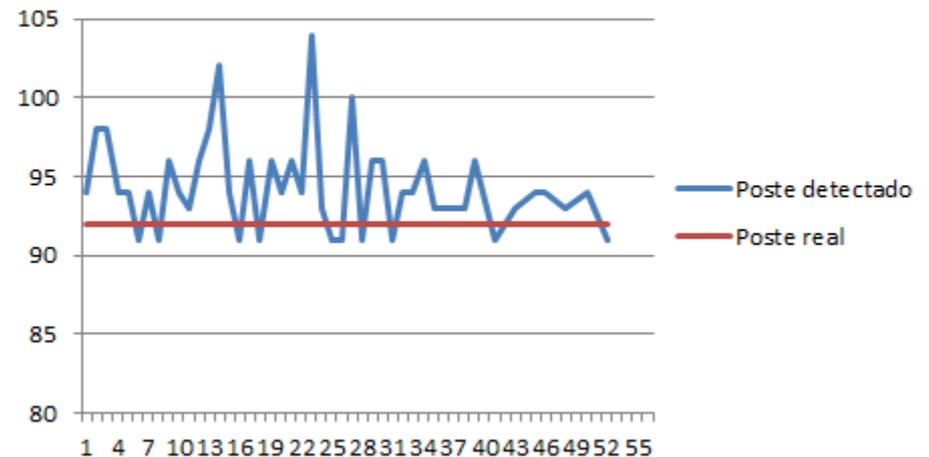


Pruebas realizadas

Ángulo al Poste



Distancia al Poste



Ventajas y desventajas

Ventajas:

- Solución fácil de mantener y extender
- Bajo consumo de procesador
- Moderado consumo de memoria (12%)
- Ángulo de objetos robusto
- Independencia de tipos de mensajes
- Acceso a visión de otros agentes

Ventajas y desventajas

Desventajas:

- Es necesaria proporcionar buena calibración
- Es necesario conocer orientación de cámara
- Se debe enfocar la base de los objetos
- Posible solapamiento de mensajes
- *Overflow* de *buffers* de mensajes
- Sensible a errores en datos de entrada

Trabajo a futuro

Localización del agente:

- Localización por Monte Carlo (Martín Llofriú, Federico Andrade).

Cooperación:

- Interpolación de objetos no identificados a partir de mensajes recibidos.