

- Búsqueda por Comunicación y Grupo Temático
- Búsqueda por Autor
- Comités (Créditos Publicación)
- Patrocinadores

5 · 6 · 7 de Septiembre · HUELVA
XXVIII Jornadas de Automática

Salir



CEA
comité
español de
automática



Universidad
de Huelva



SEGUIMIENTO DE RUTAS MULTI-ROBOT USANDO PCA INCREMENTAL

Luis Payá, Oscar Reinoso, Arturo Gil, Jose M. Pedrero, Miguel Juliá
Departamento de Ingeniería de Sistemas Industriales
Universidad Miguel Hernández. Avda. de la Universidad s/n
Ed. Torreblanca. 03202 Elche (Alicante) España
{lpaya, o.reinoso, arturo.gil, jpedrero}@umh.es

Resumen

Este trabajo presenta un método basado en apariencia aplicado al seguimiento de rutas en sistemas multi-robot, usando la información capturada por una cámara convencional y sin calibrar. En la etapa de aprendizaje, la información más relevante a lo largo de la ruta es almacenada en una base de datos creada mediante técnicas PCA incremental. Gracias a esta técnica, el robot que sigue la ruta puede comenzar la misma mientras el líder aun la está grabando. Durante la navegación, el robot seguidor lleva a cabo en primer lugar, un proceso de autocalización, comparando la vista actual con la información almacenada en la base de datos y usando técnicas probabilísticas. Tras ello, se realiza la etapa de control, en la que utilizando un regulador difuso, se calcula su velocidad lineal y angular para seguir la ruta grabada. En todo momento se trabaja con la apariencia global de las imágenes, sin necesidad de extraer puntos característicos. Los resultados experimentales muestran la robustez de los algoritmos en entornos interiores. Este método se puede aplicar en tareas colaborativas, donde un robot debe seguir a otro robot guía o en tareas de vigilancia, donde se debe repetir continuamente una trayectoria dentro de un edificio.

Palabras Clave: Visión por Computador, PCA Incremental, Localización Probabilística, Métodos Basados en Apariencia, Lógica difusa.

1 INTRODUCCIÓN

Uno de los problemas habituales que se presenta en robótica colaborativa implica el seguimiento de una ruta a lo largo de un entorno industrial o de oficinas para cumplir ciertas tareas tales como vigilancia, reparto o ensamblaje. Asimismo, el conocido problema de control de formaciones se puede estudiar como un problema de seguimiento de rutas, donde uno o varios robots deben seguir la ruta que el líder está registrando con un cierto desfase en el

espacio y en el tiempo, con aplicaciones en los campos de búsqueda y rescate. En general, para llevar a cabo este tipo de navegación desde un punto a otro dentro del entorno, se requiere un mapa. Recientemente, se están realizando grandes esfuerzos investigadores sobre el problema de SLAM (Simultaneous Localization And Mapping), usando información visual. Estos métodos tratan de construir un mapa global del entorno determinando simultáneamente la localización del robot en dicho mapa. Habitualmente, estas técnicas confían en la extracción de marcas o puntos característicos de las escenas [3], [12].

Sin embargo, para resolver el problema del seguimiento de rutas, no es necesario construir ni utilizar mapas extremadamente sofisticados para controlar los movimientos del robot seguidor. Basta con una etapa de aprendizaje, en la que se crea la base de datos con información de la ruta a seguir y una etapa de navegación, en la cual, el segundo robot sigue la ruta comparando la información sensorial actual con los datos almacenados. Los métodos clásicos realizan este control basándose en el concepto de “visual servoing”, donde se utiliza el Jacobiano de la imagen, que relaciona el cambio en las coordenadas de la imagen con los cambios en el movimiento del robot. Dicho Jacobiano se obtiene habitualmente a partir de los datos de posición de varias marcas del entorno [2]. Sin embargo, otro tipo de investigaciones sugieren que estos procesos se podrían llevar a cabo comparando la información visual general de las imágenes, sin necesidad de extraer ninguna característica, lo cual sería muy útil para escenas complicadas del mundo real en las que extraer modelos adecuados para reconocimiento puede ser muy complicado. P. ej, [8] desarrolla un método consistente en la comparación directa de imágenes de baja resolución. [14] hace uso del histograma color para llevar a cabo la autocalización, trabajando con otras características como textura y densidad de bordes. Asimismo, es posible reducir la complejidad del problema trabajando en el subespacio PCA (Principal Components Analysis), como en [5] o en [6], donde se utilizan técnicas PCA para crear la base de datos, y

técnicas probabilísticas para la localización dentro de esa base de datos. Por último [7] trata de reducir la información de las imágenes omnidireccionales capturadas por el robot utilizando la transformada de Fourier.

En el presente trabajo, se presenta un método basado en apariencia para el seguimiento de rutas, donde se ha utilizado PCA incremental para construir la base de datos, técnicas probabilísticas para la localización del robot y lógica difusa para el control durante la fase de navegación. En primer lugar, se presenta el formato de representación del entorno utilizado. A continuación, se muestran los fundamentos de la localización y control durante el seguimiento de la ruta. Para finalizar, se presentan los resultados y conclusiones del trabajo.

2 CONSTRUCCIÓN DE LA BASE DE DATOS MEDIANTE PCA INCREMENTAL

2.1 PCA

Uno de los problemas que surgen cuando se trabaja con métodos basados en apariencia es el tamaño de la base de datos. Debido a que no se trabaja con características extraídas sino con la apariencia de la imagen completa, la dimensión de los datos obtenidos con las cámaras es muy elevada. En nuestro caso, trabajar con imágenes 64x64 supone tener vectores de datos en un espacio de 4096 dimensiones. Sin embargo, todos estos datos son generados mediante un proceso que tiene únicamente 3 grados de libertad (posición y orientación del robot). De este modo, conviene llevar a cabo una reducción de la dimensionalidad de los datos antes de almacenar las imágenes. Dado que los píxeles de las imágenes tienden a estar muy correlados, una fase previa de reducción natural consiste en un análisis PCA (Principal Components Análisis), como se muestra en [4] y en [9]. Cada imagen $\bar{x}_j \in \mathfrak{R}^{M \times 1}$, $j=1 \dots N$, siendo M el número de píxeles y N el número de imágenes, puede ser transformada en un vector de características (también denominado proyección de la imagen) $\bar{p}_j \in \mathfrak{R}^{K \times 1}$; $j=1 \dots N$, compuesto por las K características PCA que contienen la información más relevante de la imagen, con $K \leq N$. En los métodos de PCA tradicionales, el primer paso consiste en construir la matriz de datos, que contiene las imágenes del entorno. La transformación PCA se calcula a partir de la matriz de covarianza de la matriz de datos, usando SVD y el método de Turk y Pentland [13], y da lugar a una transformación lineal que aplicada a la matriz de datos, proporciona una nueva matriz que contiene la información más relevante. La matriz de

transformación está compuesta por los autovectores ordenados en columnas. Dependiendo del número de autovectores que se tome, se reducirá en mayor o menor medida la dimensión de los datos. La figura 1 muestra como se pueden reducir dimensionalmente las imágenes de la ruta usando PCA (en la figura, cada imagen es reducida a $K = 3$ dimensiones, tomando una matriz de transformación con 3 autovectores). En la figura 2 se muestra la información almacenada en cada una de las componentes PCA de una de las imágenes de la ruta.

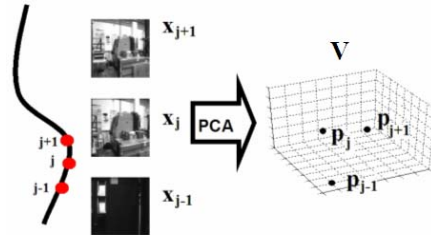


Figura 1: Mediante la utilización de PCA, la información de las escenas de la ruta se reduce dimensionalmente.

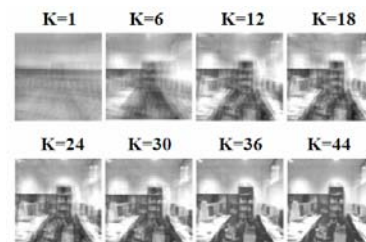


Figura 2: Reconstrucción de una escena de una ruta de 45 puntos, utilizando diferente número de autovectores. La reconstrucción para $K=44$ corresponde a la imagen original 64x64.

Sin embargo, para construir la base de datos mediante este método, es necesario que todas las imágenes estén disponibles antes de llevar a cabo la compresión, de modo que ningún robot de los que deben seguir la ruta puede iniciar el recorrido hasta que el líder la haya finalizado. Para superar este inconveniente, se ha optado por un modelo de PCA incremental que permite construir la base de datos progresivamente; según van llegando imágenes, se van incorporando a la base de datos, y se van actualizando las proyecciones existentes y la matriz de transformación PCA. De este modo, en cada iteración, tendremos un conjunto de observaciones del entorno (vectores PCA) desde un conjunto de puntos situados sobre la ruta a seguir.

2.2 PCA Incremental

Hasta el momento, se han propuesto varios algoritmos para llevar a cabo PCA de modo incremental [1]. Se puede demostrar como, cuando se

dispone de un conjunto de autovectores de un conjunto de imágenes, al añadir una nueva imagen a la base de datos, estos autovectores y la proyección de las imágenes previas pueden ser actualizados mediante el siguiente algoritmo de cuatro pasos:

1. En primer lugar, se actualiza la media:

$$\bar{m}' = \frac{1}{N+1} (N \cdot \bar{m} + \bar{x}_{N+1}) \quad (1)$$

2. A continuación, el conjunto de autovectores debe ser actualizado para que incluya la información de la nueva imagen \bar{x}_{N+1} . Para hacer esto, se calcula el vector residual, que es la diferencia entre la reconstrucción y la imagen original $N+1$: $\bar{h}_{N+1} = (V \cdot \bar{p}_{N+1} + \bar{m}) - \bar{x}_{N+1}$. Este vector es ortogonal a los autovectores antiguos. A continuación, se normaliza dividiendo por su módulo, obteniendo \hat{h}_{N+1} .

3. La nueva matriz de autovectores V' se puede obtener añadiendo \hat{h}_{N+1} a V y rotando la matriz resultante, de acuerdo con la siguiente expresión:

$$V' = \left[V \mid \hat{h}_{N+1} \right] \cdot R \quad (2)$$

Donde R es la solución al problema de diagonalización $D \cdot R = R \cdot \Lambda'$ y D :

$$D = \frac{N}{N+1} \cdot \begin{bmatrix} \Lambda & \bar{0} \\ \bar{0}^T & 0 \end{bmatrix} + \frac{N}{(N+1)^2} \cdot \begin{bmatrix} \bar{p} \cdot \bar{p}^T & \delta \cdot \bar{p} \\ \delta \cdot \bar{p}^T & \delta^2 \end{bmatrix} \quad (3)$$

Siendo $\delta = \bar{h}_{N+1} \cdot (\bar{x}_{N+1} - \bar{m})$, $\bar{p} = V^T \cdot (\bar{x}_{N+1} - \bar{m})$ y Λ una matriz diagonal que contiene los autovalores originales. De este modo, si $V \in \mathfrak{R}^{M \times K}$, entonces $V' \in \mathfrak{R}^{M \times (K+1)}$. En este punto, se debe estudiar si esta nueva dimensión aporta información significativa o no.

4. Para finalizar, las representaciones de las imágenes se actualizan mediante la siguiente expresión:

$$\bar{p}_{i(N+1)} = (R')^T \cdot \begin{bmatrix} \bar{p}_{i(N)} \\ 0 \end{bmatrix} + [V \mid \hat{h}_{N+1}]^T \cdot (\bar{m} - \bar{m}') \quad (4)$$

Este algoritmo se debe ejecutar cada vez que el robot líder captura una nueva imagen de la ruta. La figura 3 muestra cuales son los datos de entrada y de salida en cada iteración del algoritmo PCA incremental.

En otros trabajos [10] se ha mostrado la eficiencia y los detalles de implementación de este método incremental para navegación de robots.

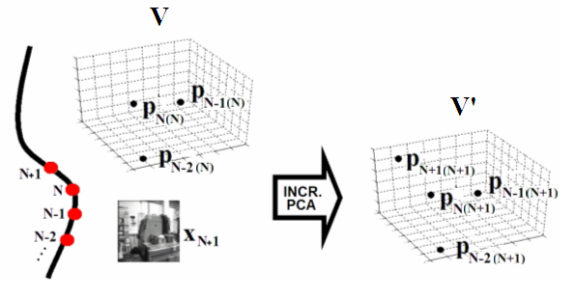


Figura 3: Datos de entrada y de salida en cada instante en un método PCA incremental con $K=3$ dimensiones.

3 LOCALIZACIÓN Y CONTROL PARA EL SEGUIMIENTO DE RUTAS.

El robot que sigue la ruta pregrabada debe ejecutar dos tareas sucesivamente: auto-localización y control.

3.1 AUTO-LOCALIZACIÓN

El robot captura una imagen y, usando esta información, debe conocer cual de las observaciones de la base de datos es la más cercana. Para ello, se debe proyectar la imagen actual en el eigenspacio calculado por el líder. El resultado será un vector de K componentes que contiene la información principal de la escena. A continuación, este vector será comparado con los almacenados en la base de datos. El que ofrece la mínima distancia Euclídea es el correspondiente a la posición actual del robot.

Sin embargo, en entornos de oficinas, este método sencillo para localización tiende a fallar a menudo como consecuencia del fenómeno de “*visual aliasing*”, consistente en que la información visual capturada desde dos localizaciones distantes entre ellas puede ser muy similar, debido a la estructura repetitiva del entorno. Para evitar estos problemas, se ha implementado un método probabilístico, basado en un proceso de Markov. La posición del robot en cada momento se puede estimar aplicando la regla de Bayes:

$$p(x|z; \theta) \propto p(z|x; \theta) \cdot p(x) \quad (5)$$

Donde $p(x)$ es la probabilidad de que el robot se encuentre en la posición x antes de observar z . Este valor se estima usando la información previa y el modelo de movimiento. $p(z|x)$ es la probabilidad de observar z si la posición del robot es x . De este modo, se debe deducir un método que permita estimar el modelo de observación. En este trabajo, la distribución $p(z|x)$ se modela a través de una suma de

kernels gaussianos, centrados en los k puntos más similares de la ruta:

$$p(z|x) = \frac{1}{k} \cdot \sum_{i=1}^k \left(\gamma_j \cdot e^{-\left(\frac{x-x_j}{\sigma}\right)^2} \right); \quad j=1 \dots N \quad (6)$$

Cada kernel se pondera por un valor de confianza $\gamma_j \in [0,1]$, que depende del grado de similitud de la proyección de la imagen actual con las proyecciones de la base de datos.

$$\gamma_j = \frac{(d_{\max}/d_j)-1}{(d_{\max}/d_{\min})-1}; \quad \gamma_j \in [0,1]; \quad j=1 \dots N \quad (7)$$

A continuación, se aplica el modelo de movimiento, modelizado mediante una segunda función gaussiana a partir de la posición y velocidad previas. Al fin, se suma en cada punto la contribución de cada kernel resultante, c_j , y para finalizar, se toma como nueva posición el punto con mayor contribución c_j^{\max} . En las figuras 4 y 5 se muestra este proceso para $k=5$ kernels.



Figura 4: Distancia de la imagen actual a las almacenadas en la BBDD, y localizaciones más probables.

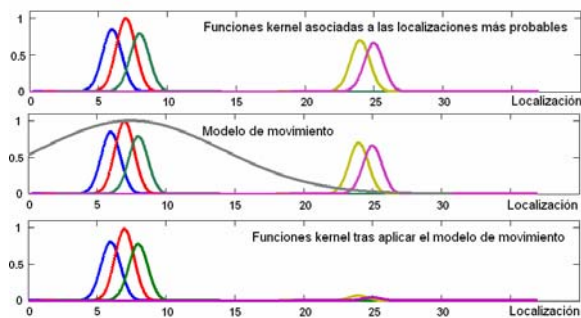


Figura 5: Cálculo de $p(x/z)$ mediante un método probabilístico, utilizando kernels gaussianos.

En primer lugar, tal y como se muestra en la figura 4, se seleccionan las 5 localizaciones más cercanas a la imagen actual. En esta figura se observa un caso de

“visual aliasing” puesto que las imágenes candidatas proceden de dos zonas del entorno distantes entre ellas. A continuación, en la figura 5 se muestran los 5 kernels en las 5 posiciones seleccionadas, ponderado cada uno de ellos por un factor que depende de las distancias calculadas. A continuación, se aplica el modelo de movimiento, resultando al fin los 5 kernels de la gráfica inferior de la figura 5. Sumando la contribución de estos kernels en cada punto, podremos deducir la localización más probable a partir de toda la información de que se disponía.

Sin embargo, este método funciona bien únicamente si se dispone de una estimación robusta de la posición inicial del robot. De este modo, el comienzo de la navegación podría ser un problema si el robot se encuentra lejos de la ruta. Para resolver este problema, se utilizará un método de clustering. El robot realiza pequeños movimientos lineales y angulares alrededor de la posición inicial, capturando imágenes durante el movimiento. Cada imagen es localizada a través de sus proyecciones PCA comparándola con todas las almacenadas en la base de datos. Entonces, cada nueva imagen se clasifica en el grupo (cluster) cuyo centro es más cercano a la localización de la imagen. Si esta distancia está por encima de cierto umbral, se constituye un nuevo cluster. De lo contrario, será incluida en el cluster correspondiente y su centro será actualizado. Una vez que todas las imágenes están clasificadas, los grupos con menor número de imágenes son descartados y se selecciona el grupo en el cual la varianza de las posiciones de los elementos es menor. La posición inicial es estimada entonces como el centro de dicho cluster. La figura 3 muestra este método. En este caso, el cluster 3 sería eliminado y se seleccionará el cluster 2 y la posición 9.

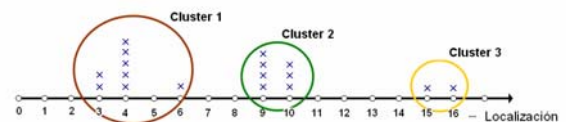


Figura 6: Método de agrupación para estimar la posición inicial del robot seguidor.

3.2 CONTROL

Para cada una de las imágenes de la base de datos, j , se obtiene un conjunto de N' subventanas $\tilde{w}_j^i \in \mathbb{R}^{M' \times I}$.

Estas subventanas se obtienen rastreando la imagen original con un incremento en el eje horizontal (Fig. 7). A continuación, se lleva a cabo un proceso de compresión PCA, calculando los componentes PCA de cada subventana $\tilde{f}_j^i \in \mathbb{R}^{K' \times I}$, donde $K' \leq N'$. La figura 7 muestra estas proyecciones como puntos negros en el caso $K'=3$. Durante la navegación

autónoma, se toman tres subventanas ($\tilde{w}_j^A, \tilde{w}_j^B, \tilde{w}_j^C$) sobre la imagen capturada actualmente (figura 7) y se rastrean sobre la banda central de la imagen correspondiente. Para hacer esto, una vez que el robot conoce su localización, se calculan las componentes PCA de estas tres subventanas. Esta operación devuelve tres vectores de K' componentes ($\tilde{f}_j^A, \tilde{f}_j^B, \tilde{f}_j^C$), marcados como cruces rojas en la figura 7. Entonces, se extraen las proyecciones más similares a cada una de ellas (dentro de las esferas dibujadas). A continuación, se calculan las velocidades lineal y angular utilizando un controlador difuso, cuyas entradas son las proyecciones más similares a cada una de las tres subventanas. El regulador analiza estos datos y tras resolver posibles inconsistencias, devuelve las velocidades lineal y angular necesarias para tender a la ruta.

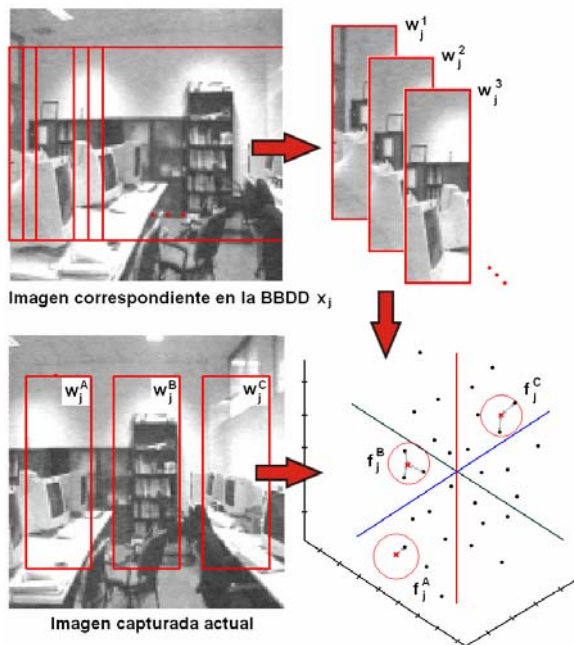


Figura 7: Obtención de los datos para el cálculo de la velocidad lineal y angular del robot seguidor.

4 RESULTADOS

4.1 EXPERIMENTOS REALIZADOS

Para testar los algoritmos expuestos, se han utilizado dos robots Pioneer 3-AT equipados con una cámara cada uno de ellos y con procesadores P_A y P_B a bordo. R_A es el robot líder, que está grabando la ruta, y R_B es el robot seguidor. Además, se ha utilizado un procesador independiente P_C . Todos los computadores se comunican mediante una arquitectura basada en el estándar CORBA.

Para empezar, el robot A es teleoperado a través de la ruta deseada. Después de unos instantes, el robot B comienza su navegación. Las tareas que cada procesador lleva a cabo son:

Procesador P_A :

- Lee una nueva imagen \tilde{x}_{N+1} , la compara con la imagen anterior \tilde{x}_N y la almacena si la correlación está por debajo de cierto umbral.
- Actualiza el espacio PCA V' . Calcula la proyección de la imagen actual, obteniendo \tilde{p}_{N+1} . Borra todos los datos anteriores.
- Crea el subespacio PCA con las subventanas U_j y calcula las proyecciones f_j^i .
- Envía V' , \tilde{p}_{N+1} , U_j y f_j^i a P_C .

Procesador P_C :

- Recalcula las proyecciones de las imágenes: $\tilde{p}_{i(N+1)}$.
- Calcula el error de reconstrucción usando K y $K+1$ autovectores. Si está por debajo de cierto umbral, se reduce la dimensión del espacio PCA de $K+1$ a K de nuevo.
- Se envían a P_B las matrices V' , U_j y las proyecciones actualizadas $\tilde{p}_{i(N+1)}$ y f_j^i .

Procesador P_B :

- Captura una nueva imagen y la proyecta en V' .
- Calcula la posición actual del robot B utilizando el método probabilístico expuesto.
- Calcula la acción de control a aplicar a R_B utilizando el controlador difuso.

Utilizando esta distribución de tareas, se han llevado a cabo varios experimentos para demostrar la validez del método descrito, utilizando dos robots Pioneer 3-AT. El objetivo final de los experimentos desarrollados, era comprobar la validez del método PCA incremental, comparándolo con el método PCA tradicional. Asimismo, se pretendía comprobar la robustez del método de localización probabilístico, así como la influencia del número de autovectores. Los resultados obtenidos se muestran en el siguiente apartado.

4.2 RESULTADOS OBTENIDOS

La figura 8 muestra una ruta típica, grabada en un entorno interior y la trayectoria seguida por el segundo robot. El robot seguidor es capaz de localizarse correctamente al principio (a pesar de encontrarse relativamente lejos de la ruta), tender a la misma y seguirla, mostrando un comportamiento muy bueno en los tramos rectos y un error aceptable

en las curvas. En general, el comportamiento del robot seguidor es muy estable hasta el final del recorrido.

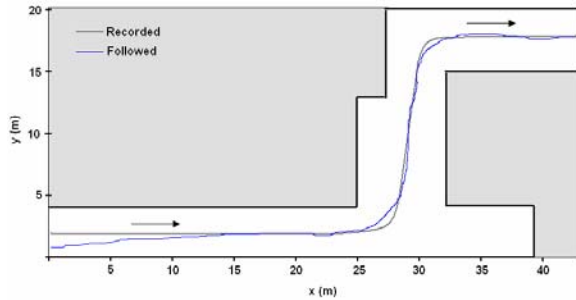


Figura 8: Resultados de la navegación. Ruta grabada y seguida para $K=20$, usando PCA incremental, en un entorno interior de oficinas.

La figura 9 muestra la evolución de la localización durante la navegación del segundo robot y la probabilidad calculada en cada iteración, la cual puede ser una buena medida de la precisión durante la navegación. La localización muestra una evolución correcta (a pesar del fenómeno de “visual aliasing” en entornos de oficinas) y el robot se recupera correctamente de errores puntuales en la localización (como los que se producen en las iteraciones 100 y 170). Asimismo, la probabilidad comienza tomando valores pequeños (al principio, el robot se encuentra relativamente lejos de la ruta) y tiende a incrementarse cuando se acerca al tramo recto, decreyéndose de nuevo en las zonas curvas.

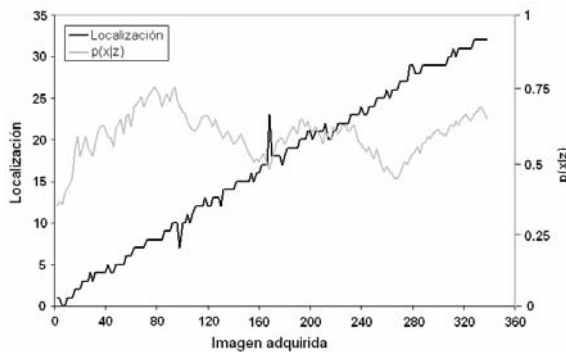


Figura 9: Evolución de la localización actual y probabilidad durante la navegación para el ejemplo de la figura 8.

La figura 10 es el resultado de varios experimentos llevados a cabo con diferentes números de autovectores. Se aprecia como la navegación es óptima para un número medio de autovectores. Si K es demasiado pequeña, la información contenida en las proyecciones es demasiado pobre para tener una buena navegación, y si K es demasiado grande, crece el coste computacional, con lo cual, la acción de

control se refresca menos frecuentemente dando lugar a mayores errores de navegación.

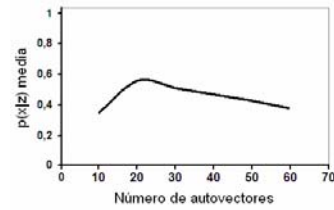


Figura 10: Probabilidad media durante toda la navegación para diferente número de autovectores.

Si se compara el método PCA tradicional con el PCA incremental, el método tradicional funciona ligeramente mejor en cuanto al error medio en el seguimiento de la ruta. Sin embargo, PCA incremental lleva a cabo la tarea correctamente, con las ventajas que ello supone. La figura 11 muestra los errores de navegación obtenidos con ambos métodos para diferentes cantidades de autovectores.

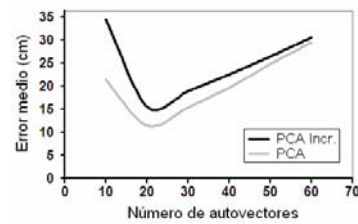


Figura 11: Error medio entre las rutas pregrabada y seguida, para diferente número de autovectores, comparando los métodos PCA y PCA incremental.

5 CONCLUSIONES

En este trabajo se ha presentado un esquema para el seguimiento visual de rutas multi-robot mediante un método basado en apariencia. La solución propuesta utiliza imágenes de baja resolución tomadas por una cámara de video convencional y técnicas PCA para extraer la información más relevante de las escenas capturadas a lo largo de la trayectoria. Asimismo, para permitir que un segundo robot pueda seguir la ruta del robot líder mientras éste aun está grabándola, se ha utilizado un algoritmo de PCA incremental.

El objetivo final del trabajo es que un robot pueda seguir la trayectoria de otro robot líder con un desfase (en el espacio o en el tiempo). Para ello, se ha implementado en primer lugar un algoritmo probabilístico que permite calcular su posición actual, evitando los errores debidos al “visual aliasing” y a continuación se ha utilizado un regulador difuso, también basado en la apariencia de las escenas, que permite detectar desfases

horizontales entre la posición actual del robot y la correspondiente en la base de datos y devuelve las velocidades lineal y angular que se deben proporcionar al robot para que tienda a la ruta y la siga hasta el final.

Para finalizar, se han realizado diversos experimentos con dos robots Pioneer 3-AT. Dichos experimentos muestran como el proceso empleado permite el seguimiento de rutas de una forma precisa y robusta. Ahora estamos trabajando en otros métodos de control que permitan reducir el error durante la navegación, teniendo en cuenta otros efectos como los cambios de iluminación del entorno y las posibles oclusiones.

Agradecimientos

Este trabajo ha sido subvencionado por el Ministerio de Educación y Ciencia a través del proyecto DPI2004-07433-C02-01 ‘Herramientas de teleoperación colaborativa. Aplicación al control cooperativo de robots’, y por la Fundación Quórum: Parque científico y empresarial de la Universidad Miguel Hernández, proyecto PCT-G54016977-2005 ‘Robots cooperativos para la vigilancia e inspección de edificios e instalaciones industriales’.

Referencias

- [1] Artac, M., Jogan, M., Leonardis, A., (2002) “Mobile Robot Localization Using an Incremental Eigenspace Model”, *Proc. IEEE Int. Conf. on robotics and Automation*, pp. 1205-1030.
- [2] Burschka, D., Hager, G., (2001) “Vision-Based Control of Mobile Robots”, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1707-1713.
- [3] Fenwick, J., Newman, P., Leonard, J., (2002) “Cooperative Concurrent Mapping and Localization”, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.1810-1817.
- [4] Kirby, M., (2001) Geometric data analysis. Wiley Interscience.
- [5] Kröse, B., Bunschoten, R., Hagen, S., Terwijn, B., Vlassis, N., (2004) “Household robots: Look and learn”, *IEEE Robotics & Automation magazine*, 11(4), pp. 45-52.
- [6] Maeda, S., Kuno, Y., Shirai, Y., (1997) “Active navigation vision based on eigenspace analysis”, *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 1018-1023.
- [7] Menegatti, E., Maeda, T., Ishiguro, H., (2004) “Image-based memory for robot navigation using properties of omnidirectional images”, *Robotics and Autonomous Systems*, 47(4), pp. 251-267.
- [8] Matsumoto, Y., Ikeda, K., Inaba, M., Inoue, H., (1999) “Visual Navigation Using Omnidirectional View Sequence”, *Proc. IEEE Conf. on Intelligent Robots and Systems*, pp. 317-322.
- [9] Nayar, S.K., Nene, N.A., Murase, H., (1996) “Subspace methods for robot vision”, *IEEE Transactions on Robotics and Automation*, 12(5), pp. 750-758.
- [10] Payá, L., Reinoso, O., Gil, A., Pedrero, J., Ballesta, M., (2007) “Appearance-Based Multi-Robot Following Routes Using Incremental PCA”, *Futura publicación, Int. Conf. on Knowledge-Based and Intelligent Information and Engineering Systems*.
- [11] Payá, L., Vicente, M.A., Navarro, L., Reinoso, O., Fernández, C., Gil, A., (2005) “Continuous navigation of a mobile robot with an appearance-based method”, *Proc. 2nd Int. Conf. on Informatics in Control, Automation and Robotics*, pp. 443-446.
- [12] Thrun, S., (2002) “Robotic Mapping: A Survey”, *Exploring Artificial Intelligence in the New Milenium*, Morgan Kaufmann.
- [13] Turk, M., Pentland, A., (1991) “Eigenfaces for recognition”, *Journal on Cognitive Neuroscience*, 3(1), pp. 71-86.
- [14] Zou, C., Wei, T., Tan, T., (2003) “Mobile robot self-localization based on global visual appearance features”, *Proc. Of the IEEE Int. Conf. on Robotics and Automation*, pp. 1271-1276.