

NAVEGACIÓN CONTÍNUA DE UN ROBOT MÓVIL BASADA EN APARIENCIA

L. Payá, M.A. Vicente, O. Reinoso, C. Fernández, L.M. Jiménez
Universidad Miguel Hernández, Div. Ingeniería de Sistemas y Automática
Departamento de Ingeniería de Sistemas Industriales, Campus de Elche
Avda. de la Universidad s/n 03202 Elche (Alicante), España.
lpaya, suni, o.reinoso, c.fernandez, luis.jimenez@umh.es

Resumen

El presente trabajo expone la aplicación de los métodos basados en apariencia a la navegación de un robot móvil. Estos métodos se basan en la comparación directa de imágenes sin realizar ninguna extracción de características previa, presentando varias ventajas respecto a los métodos basados en el modelo, como su aplicación a entornos no estructurados y la relativa sencillez del control a implementar. Sin embargo, el mayor inconveniente de estas técnicas es que se requieren grandes memorias y un alto coste computacional puesto que se basan en la comparación continua de las imágenes actuales con todas las almacenadas en una base de datos. De este modo, uno de los puntos clave para su éxito, es la naturaleza y la cantidad de información sobre el entorno que debe ser almacenada para hacer posible la navegación. Este trabajo presenta dos alternativas aplicadas al seguimiento de rutas previamente aprendidas. El primero de ellos consiste en almacenar imágenes frontales de baja resolución a lo largo de la ruta a seguir, adquiriéndolas con un par de cámaras paralelas. Bajo este sistema, se han probado varios esquemas de control para optimizar la precisión durante la navegación, tales como control P, PD y PD con parámetros variables. La segunda alternativa consiste en la reducción de la dimensión de los datos a almacenar, calculando previamente la información más relevante de cada imagen. En este trabajo se expone como se puede llevar a cabo esta reducción utilizando el método de PCA. Este método permite una clara reducción del coste computacional sin necesidad de disminuir la resolución de las imágenes, lo cual implica que puede ser usado en entornos extremadamente no-estructurados y sin que ello suponga una reducción de la velocidad media del robot durante su navegación.

Palabras clave: Aprendizaje automático, navegación continua, método basado en apariencia, PCA, VSR

1 INTRODUCCIÓN

Para poder llevar a cabo la navegación de un robot móvil desde un punto a otro en un entorno, se requiere un mapa del mismo. Dicho mapa debe contener información suficiente para hacer posible que el robot reconozca su posición actual dentro del entorno y para calcular la acción de control necesaria que le permita navegar desde el punto origen al destino, siguiendo una trayectoria predeterminada.

La investigación clásica sobre robots móviles se ha centrado en métodos que utilizan modelos geométricos para llevar a cabo la auto-localización y navegación [6][16]. Estas técnicas utilizan marcas naturales o artificiales de una escena como referencias para guiar al robot a lo largo de la ruta deseada. El reconocimiento de patrones se consigue comparando características de la imagen de entrada con características que han sido almacenadas previamente. Estas técnicas suponen una gran complejidad debido a la dificultad en la extracción de características y comparación de patrones en entornos reales y cambiantes. Además, pueden surgir problemas adicionales en entornos no estructurados, en los cuales no se pueden colocar marcas artificiales y no existen marcas naturales que puedan ser segmentadas con precisión. En este caso, puede resultar más conveniente utilizar las propiedades de la apariencia global de las imágenes.

Esta aproximación basada en la apariencia ha atraído recientemente el interés de investigadores en el campo de la robótica y visión por computador. Las imágenes son memorizadas sin ninguna extracción de características previa, y el reconocimiento se consigue mediante la comparación directa de imágenes. De este modo, se espera que este método sea útil en escenas complejas en el mundo real, donde es difícil crear modelos adecuados para reconocimiento.

El punto común de estas técnicas es que constan de dos fases, la de aprendizaje, en la cual el robot almacena información visual general desde varios puntos de vista dentro del entorno de interés, y la fase de navegación autónoma, en la que

el robot lleva a cabo la navegación comparando la información visual actual con la previamente almacenada.

La principal desventaja de estas técnicas es que requieren grandes cantidades de memoria para almacenar la información necesaria del entorno y que suponen un alto coste computacional para realizar las comparaciones durante la navegación. De este modo, los puntos clave de estos métodos residen en el tipo y cantidad de información a almacenar y en como llevar a cabo la comparación entre la vista actual y las almacenadas para reducir el coste computacional.

Los investigadores han propuesto varios métodos para realizar la auto-localización y navegación basada en la apariencia global. Matsumoto et al. presentan el método VSRR (View-Sequence Route-Representation) [9], que consiste en la comparación directa de imágenes de baja resolución. Jones et al. proponen un método basado en el mismo concepto pero usando un par de cámaras e información odométrica [3]. Otros métodos hacen uso del histograma color para llevar a cabo la localización. El problema es que el histograma no contiene información espacial, de modo que es necesario extraer otras características como la textura, densidad de bordes, etc. [19][21]. Regini et al. [15] proponen un método que calcula las relaciones espaciales entre las regiones de color. Además la complejidad del problema puede reducirse trabajando en el subespacio PCA [1][8] o intentando extraer las regiones de la imagen que contienen la información más relevante [20].

En este trabajo, se presentan dos métodos para llevar a cabo la navegación utilizando únicamente información visual. El primero de ellos se basa en el modelo VSRR con un par de cámaras, en el cual, el tamaño de las imágenes a almacenar es reducido tratando de minimizar la resolución de las mismas, y el tamaño de la base de datos es optimizado utilizando una etapa de aprendizaje automatizada. El segundo método muestra como la información puede ser reducida trabajando en el subespacio PCA y como este concepto se puede extender a la navegación del robot. Además se exponen las leyes de control necesarias para gobernar la velocidad de avance y de giro del robot, utilizando varios esquemas de regulación diferentes.

El artículo se estructura del siguiente modo: La sección 2 describe los puntos clave de la aplicación desarrollada con las nuevas contribuciones relativas al almacenamiento de información y leyes de control utilizando imágenes de baja resolución. El apartado 3 estudia el uso de PCA para comprimir la información, y como

este concepto puede extenderse a la navegación del robot. Finalmente, el punto 4 expone las conclusiones y los futuros trabajos que ya se están desarrollando para mejorar estos métodos.

2 NAVEGACIÓN CONTINUA DE UN ROBOT UTILIZANDO IMÁGENES DE BAJA RESOLUCIÓN

Para desarrollar esta aplicación, se ha utilizado el robot móvil B21r, el cual posee cinemática síncrona y tracción a las cuatro ruedas. Esto significa que las cuatro ruedas pueden girar respecto al eje vertical que pasa por sus centros, permaneciendo paralelas en todo momento. La separación entre los sistemas de tracción y dirección hace posible controlar independientemente la velocidad de avance v , y la velocidad de giro ω . Estas dos variables serán las utilizadas para controlar el movimiento del robot. Sobre el robot se ha montado una unidad *pan-tilt* con un par de cámaras Sony XC999 con sus ejes ópticos alineados. La máxima resolución de estas cámaras es de 640x480 píxeles, pero se ha implementado un servidor de imágenes que proporciona una resolución menor. El uso simultáneo de dos cámaras hará este método más robusto.

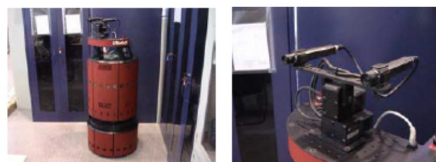


Figura 1: Material usado para desarrollar la aplicación: Robot B21r y cámaras Sony XC999

El método propuesto se basa en el modelo VSRR [9], que trata de resolver el problema de la necesidad de grandes memorias utilizando imágenes de baja resolución para modelar la ruta. Experimentos previos con resoluciones 32x32, 64x64 y 128x128 han mostrado que la resolución 32x32 proporciona resultados buenos durante la fase de navegación debido al menor coste computacional que supone, lo cual implica que el robot puede navegar a una velocidad mayor. De todos modos, el error durante el seguimiento de la trayectoria pregrabada es mayor en este caso [14]. El propósito del trabajo es el seguimiento de rutas aprendidas previamente. Para ello, se deben implementar dos fases, la de aprendizaje, en la cual se almacena información visual sobre la ruta a seguir, y la fase de navegación autónoma, en la que el robot estima su posición actual y es controlado

de manera que tienda a la ruta aprendida.

2.1 ETAPA DE APRENDIZAJE

En esta fase, el robot es guiado de forma teleoperada a lo largo de la ruta a aprender, tomando varias imágenes a lo largo de ella y almacenándolas en la base de datos. En nuestro trabajo previo [14], la ruta es descompuesta en segmentos rectos antes de llevar a cabo el aprendizaje, como expone [9]. Este hecho hace necesario almacenar, además de las imágenes, la acción de control cualitativa (izquierda, derecha) que el robot debe ejecutar en la intersección de dos segmentos adyacentes. De este modo, cuando el robot llega a una de estas intersecciones, tiene que detenerse y comenzar un movimiento de rotación pura.

Con el nuevo modelo, se propone un nuevo método de aprendizaje que posibilita la navegación continua, sin necesidad de almacenar información adicional aparte de las imágenes a lo largo de la ruta.

Para optimizar el tamaño de la base de datos, la etapa de aprendizaje ha sido automatizada. El robot captura imágenes simultáneamente con ambas cámaras en el primer punto de la ruta, y compara continuamente las vistas actuales con las almacenadas en la base de datos. El criterio usado es la correlación cruzada normalizada, cuyo valor máximo es 1. Cuando la correlación de las imágenes actuales respecto a las previamente almacenadas cae por debajo de cierto umbral, un nuevo par de imágenes es adquirido y almacenado en la base de datos. La Fig. 2 muestra una posible ruta con los puntos donde las imágenes han sido tomadas.

En las zonas rectas, las vistas cambian con relativa lentitud según el robot avanza, de modo que se almacenan imágenes nuevas con menor frecuencia. Cuando el robot está girando, la información cambia más rápidamente, con lo cual, las imágenes son almacenadas muy frecuentemente. Esto optimizará el tamaño de la base de datos. La Fig. 3 muestra la evolución de la correlación durante la etapa de aprendizaje.

2.2 NAVEGACIÓN AUTÓNOMA

Durante la navegación autónoma, el robot es situado en un punto cerca de la ruta aprendida. A partir de este punto, debe reconocer cuál de las posiciones almacenadas es la más cercana y moverse para tender a la ruta, siguiéndola hasta el punto final. Para conseguir esto, han sido implementados dos procesos que se ejecutan sucesivamente en esta etapa: auto-localización y

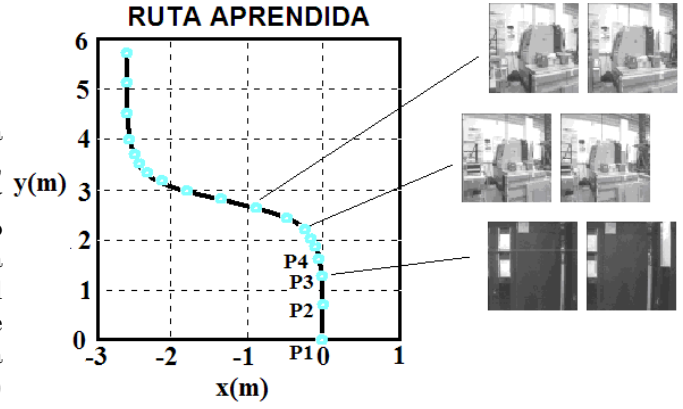


Figura 2: Base de datos creada durante la etapa de aprendizaje.

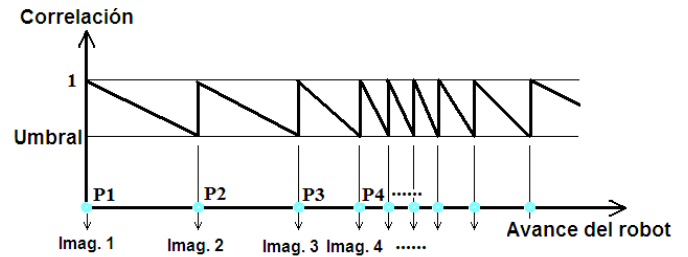


Figura 3: Evolución de la correlación durante la etapa de aprendizaje.

control.

2.2.1 Auto-Localización

Se consigue comparando las imágenes actuales completas con todas las almacenadas en la base de datos (la izquierda con todas las imágenes izquierdas y lo mismo para la imagen derecha). El criterio de comparación adoptado es la correlación cruzada normalizada:

$$\gamma_i = \frac{\sum_{x,y} [I(x,y) - \bar{I}] \cdot [A(x,y) - \bar{A}]}{E_I \cdot E_A}$$

$$E_I = \sqrt{\sum_{x,y} [I(x,y) - \bar{I}]^2}$$

$$E_A = \sqrt{\sum_{x,y} [A(x,y) - \bar{A}]^2} \quad (1)$$

I y A son las dos imágenes a comparar, cuyos valores medios son \bar{I} y \bar{A} , y cuyas energías son E_I y E_A . La principal ventaja que aporta el uso de este criterio es su insensibilidad a la iluminación de la escena y ante el ruido de entrada [7].

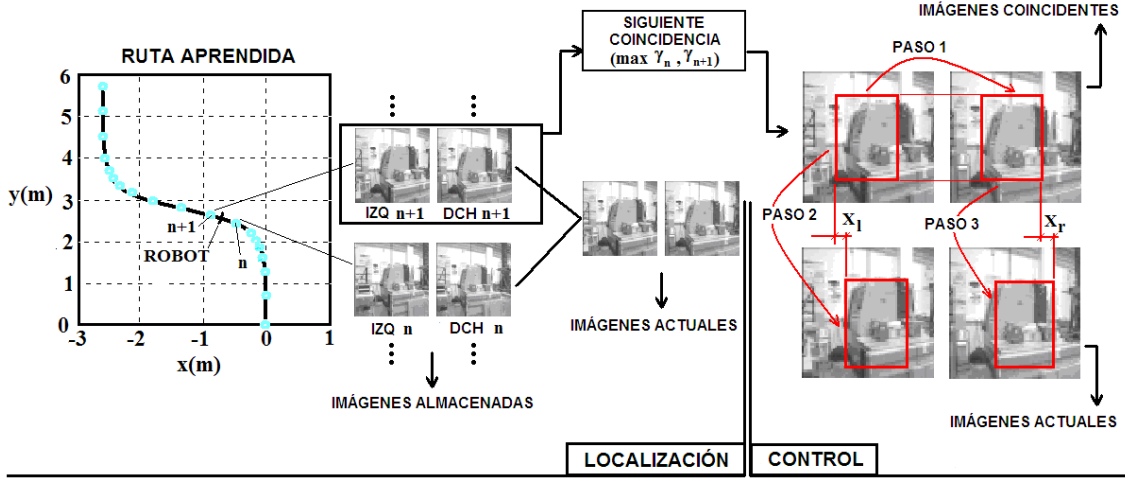


Figura 4: Tareas realizadas durante la etapa de navegación autónoma. En primer lugar, el robot lleva a cabo la auto-localización, comparando las imágenes actuales con las coincidentes anteriores y las siguientes almacenadas en la base de datos. Una vez que tenemos una nueva coincidencia, se calcula la velocidad lineal y angular del robot en base a la correlación global y al desplazamiento relativo de dos subventanas.

La media y la energía de las imágenes almacenadas puede ser calculada en un proceso off-line, lo cual puede ahorrar coste computacional durante la navegación.

Como en cada punto tenemos dos imágenes, para obtener un dato general de comparación usamos la media aritmética de las correlaciones de las imágenes izquierda y derecha.

$$\gamma_{av} = \frac{\gamma_{left} + \gamma_{right}}{2} \quad (2)$$

Al principio de la navegación, se deben comparar las imágenes actuales con todas las almacenadas, reconociendo la posición más cercana mediante la correlación media. En las siguientes iteraciones, dicha comparación únicamente se debe llevar a cabo entre la imagen actual y la coincidente anterior y la siguiente almacenadas, dado que la navegación es continua y el robot debe pasar por todos los puntos sucesivamente. Esto implica que, una vez que el robot ha comenzado su navegación, el tiempo necesario para procesar es independiente del tamaño de la base de datos, y por tanto, de la longitud de la ruta a seguir.

2.2.2 Control

En esta tarea, el robot debe corregir su dirección con el objetivo de tender a la ruta y seguirla hasta el punto final. Para conseguir este propósito, las imágenes actuales son comparadas de forma continua con las previamente coincidentes y las siguientes almacenadas (Fig.4: Localización). Una vez que tenemos una nueva coincidencia,

tomamos una subventana en la imagen izquierda de la base de datos y rastreamos sobre la imagen derecha, identificando la subventana que mejor correlación ofrece con la izquierda (Fig. 4: Control, Paso 1. Este proceso puede realizarse off-line). Como los ejes ópticos son paralelos, el offset vertical de la subventana derecha respecto a la izquierda debe ser nulo. Entonces, se lleva a cabo el mismo proceso entre la imagen coincidente izquierda y la imagen actual izquierda (Fig 4. Control, Paso 2), y finalmente entre las imágenes derechas (Fig. 4: Control, Paso 3). Los desplazamientos horizontales x_l (entre las subventanas izquierdas) y x_r (entre las subventanas derechas) permiten calcular la velocidad de giro necesaria para que el robot tienda a la ruta.

La velocidad de avance será proporcional a la correlación media, lo cual significa que cuando el robot se encuentre lejos de la ruta aprendida, la velocidad será pequeña con objeto de permitir al robot corregir su trayectoria, pero cuando la ruta está siendo seguida con relativa precisión, se permite que el robot vaya más rápidamente. Esto supone una ley de control proporcional del siguiente modo:

$$\begin{aligned} \omega^i &= k_l \cdot x_l^i + k_r \cdot x_r^i \\ v^i &= k_2 \cdot \gamma_{av}^i \end{aligned} \quad (3)$$

Siendo k_l , k_r y k_2 tres constantes. Teniendo en cuenta los resultados obtenidos experimentalmente en trabajos anteriores [14], se han adoptado los siguientes valores para estas constantes: $k_l = k_r = 0.04$ y $k_2 = 0.6$.

La siguiente figura muestra la evolución típica de la correlación durante la navegación autónoma y la influencia del umbral de aprendizaje en la precisión.

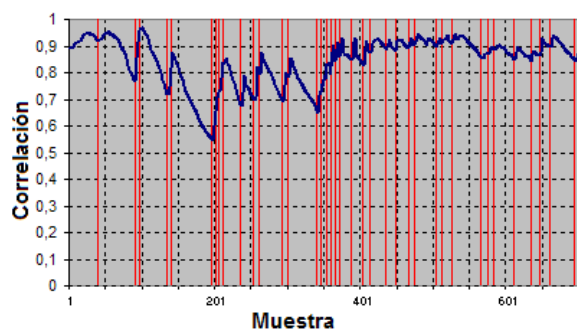


Figura 5: Evolución de la correlación durante la navegación autónoma.

Las líneas verticales indican los puntos donde las imágenes coincidentes de la base de datos cambian. Cuando esto ocurre, la correlación comienza a incrementar su valor, alcanzando un máximo cuando el robot pasa sobre el punto donde las imágenes fueron adquiridas y comienza a decrementarse hasta que las siguientes imágenes son las que más se ajustan. El valor medio de esta correlación puede ser utilizado como una medida de la precisión en el seguimiento del camino respecto al aprendizaje. Después de varios experimentos con diferentes valores para el umbral de aprendizaje, los resultados obtenidos se muestran en la Fig. 6.

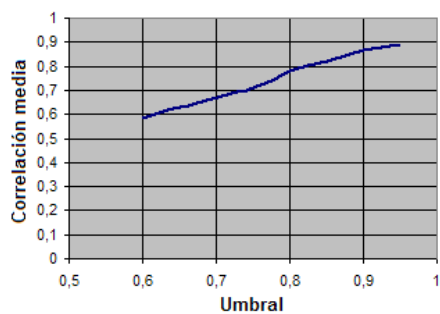


Figura 6: Correlación media durante la navegación para diferentes umbrales de aprendizaje.

Para mejorar el comportamiento del robot durante la navegación, de forma que se lleve a cabo con un mayor grado de correlación, se han testado varios esquemas de control. El segundo de ellos incluye efectos diferenciales en el control de la velocidad angular y de avance.

$$\begin{aligned} \omega^i &= k_1 \cdot [x_l^i + x_r^i] \\ &+ k_{1D} \cdot [(x_l^i - x_l^{i-1}) + (x_r^i - x_r^{i-1})] \\ v^i &= k_2 \cdot \gamma_{av}^i + k_{2D} \cdot (\gamma_{av}^i - \gamma_{av}^{i-1}) \end{aligned} \quad (4)$$

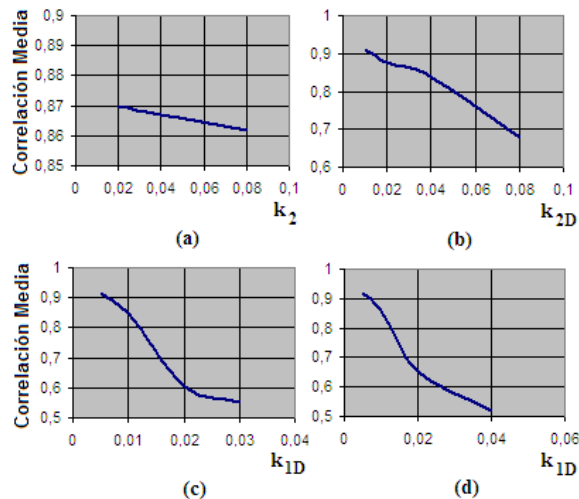


Figura 7: Correlación media durante la navegación para diferentes esquemas de control. (a) Controlador P con $k_l = k_r = 0.04$. (b) Controlador P con efecto derivativo en la velocidad de avance. $k_2 = 0.04$. (c) Controlador PD con $k_2 = 0.04$ y $k_{2D} = 0.04$. (d) Controlador PD con parámetros variables, $k_2 = 0.04$ y $k_{2D} = 0.04$.

Con este regulador se introduce un efecto de predicción del movimiento del robot respecto a la trayectoria deseada. En el caso de la velocidad lineal, cuando la correlación está creciendo, el robot se está aproximando a la ruta correctamente. En este caso, el factor derivativo es positivo, lo cual supone que el robot va más rápidamente dado que está tendiendo a la ruta correctamente. Cuando la correlación disminuye, el factor derivativo es negativo, con lo cual, el robot reduce su velocidad porque se está alejando de la ruta. Este concepto es extensible a la velocidad angular también. Esto implica que el efecto diferencial puede ayudar a mejorar la velocidad media durante la navegación y el error en el seguimiento, como se muestra en la Fig. 7. El último esquema de control con que se ha experimentado se basa en el diferencial, pero haciendo sus parámetros variables.

$$\begin{aligned} \omega^i &= k_1 \cdot [\gamma_l^i \cdot x_l^i + \gamma_r^i \cdot x_r^i] \\ &+ k_{1D} \cdot [\gamma_l^i \cdot (x_l^i - x_l^{i-1}) + \gamma_r^i \cdot (x_r^i - x_r^{i-1})] \\ v^i &= k_2 \cdot \gamma_{av}^i + k_{2D} \cdot (\gamma_{av}^i - \gamma_{av}^{i-1}) \end{aligned} \quad (5)$$

En este esquema, el efecto del desplazamiento horizontal de la subventana izquierda es multiplicado por la correlación entre las ventanas izquierdas, y lo mismo para la derecha. Dicho controlador puede ser útil cuando existe un obstáculo u oclusión que afecta únicamente a una de las cámaras, o incluso cuando existe un fallo en las mismas. En estos casos, la acción de control de la cámara que tiene el problema será multiplicada por una cantidad relativamente pequeña, con lo cual tendrá un efecto pobre sobre la navegación del robot. Además, los experimentos que han sido llevados a cabo muestran como este esquema de control mejora ligeramente los resultados que ofrece el diferencial. Estos resultados son mostrados en la Fig. 7.

3 NAVIGACIÓN CONTINUA UTILIZANDO IMÁGENES DE ALTA RESOLUCIÓN Y PCA

La principal desventaja del método de navegación antes expuesto emerge cuando las escenas utilizadas para definir la ruta son altamente no estructuradas. En ese caso, es necesario incrementar la resolución para conseguir que el método de navegación sea eficiente. Sin embargo, el utilizar imágenes de mayor resolución supone un mayor coste computacional del clasificador utilizado, lo que implica una disminución en la velocidad de navegación. En esta sección se propone un método alternativo capaz de manejar imágenes de alta resolución pero sin entlencecer la etapa de clasificación.

En el campo del reconocimiento visual de objetos basado en la apariencia se han utilizado diferentes técnicas para reducir la dimensión de los datos a tratar, permitiendo que el clasificador sólo trabaje con un conjunto reducido de atributos. El objetivo es eliminar la información irrelevante o redundante, pero manteniendo la información primordial capaz de distinguir entre patrones o clases diferentes. La técnica clásica más utilizada es el análisis de componentes principales (PCA) [13][4]. Esta técnica ha sido utilizada con éxito en el reconocimiento de objetos aplicado a tareas robóticas [10][11], y en el reconocimiento de caras humanas [5][18]. Otra técnica también muy extendida en la literatura es el análisis discriminante lineal (LDA), en este caso se trata de un método supervisado que considera las clases presentes en los datos [2][17]. La reducción de la dimensión de los datos es de hecho un proceso de extracción de características, puesto que nuevas características son extraídas de los datos originales.

Si el método de navegación propuesto en la sección anterior se intenta ejecutar utilizando imágenes de alta resolución, el coste computacional del clasificador (e.g. un k-NN con una cierta distancia métrica, como la correlación) es inaceptable. Sin embargo, es posible comprimir la información presente en este tipo de imágenes utilizando un método de extracción de características. En nuestro sistema, las imágenes originales de la ruta son comprimidas utilizando PCA en las fases antes comentadas: auto-localización y navegación autónoma.

3.1 PCA DURANTE LA ETAPA DE APRENDIZAJE

La fase de aprendizaje se realiza de forma muy similar al caso anterior de las imágenes de baja resolución. La única diferencia está en la información almacenada. Ahora, la información visual se reduce dimensionalmente mediante la técnica de PCA, de esta forma, no es necesario almacenar las imágenes completas a lo largo de la ruta, sino una versión reducida de éstas.

Los datos de entrada de la transformación PCA están en la matriz \mathbf{X} :

$$\begin{aligned} \mathbf{X} &= [\tilde{\mathbf{x}}^1 | \dots | \tilde{\mathbf{x}}^j | \dots | \tilde{\mathbf{x}}^N] \\ \tilde{\mathbf{x}}^j &= \mathbf{x}^j - \mathbf{m} \\ \mathbf{m} &= \frac{1}{N} \sum_{j=1}^N \mathbf{x}^j \end{aligned} \quad (6)$$

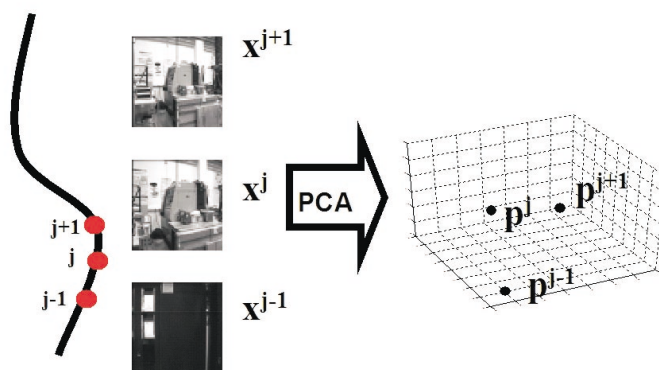


Figura 8: El tamaño original de las imágenes de la ruta se reduce mediante PCA, de ese modo se trabaja con alta resolución pero un espacio de dimensión inferior.

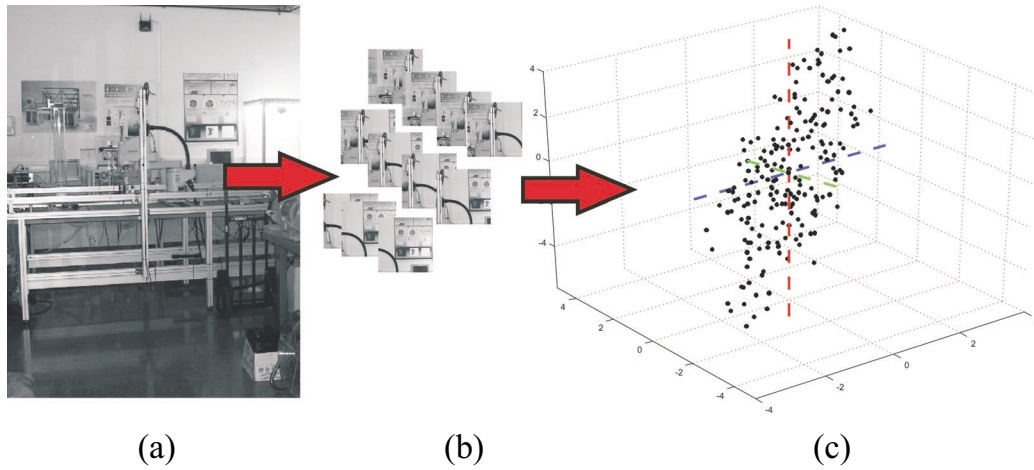


Figura 9: Compresión de las ventanas mediante PCA durante la fase de aprendizaje de la etapa de control: (a) escena original en un punto determinado de la ruta; (b) ejemplo de las ventanas utilizadas; (c) cada ventana es un vector multidimensional en el espacio imagen: la transformación de PCA está definida por el conjunto de vectores propios de la matriz de covarianza de los datos, estos vectores son las direcciones de máxima varianza en el espacio imagen original .

donde \mathbf{x}^j es cada una de las vistas representativas de la ruta (dispuestas en vectores columna) y \mathbf{m} es la imagen media del conjunto de N vistas.

La transformación PCA se calcula a partir de la matriz de covarianzas de los datos, $\mathbf{X}\mathbf{X}^T$, utilizando SVD y el truco de Turk y Pentland's [18]. Si el tamaño de la matriz de los datos es $M \times N$, donde N es el número de imágenes a lo largo de la ruta y M es el número de píxeles de cada imagen, y además, $M \gg N$, se obtendrán como máximo un total de N vectores propios. La reducción de la dimensión de los datos originales se consigue realizando el siguiente producto matricial:

$$\mathbf{P} = \mathbf{V}^T \cdot \mathbf{X} \quad (7)$$

donde \mathbf{V} es la matriz de K vectores propios en sus filas y \mathbf{P} la matriz de datos reducidos de tamaño $K \times N$, donde $K \leq N$. Al realizar la transformación anterior, las imágenes originales de M píxeles, se reducen a un vector de sólo K componentes, por tanto, la ruta estará definida por N vectores de K componentes. La figura 8 ilustra el proceso descrito en el caso que $K = 3$. De esta manera, cada una de las imágenes descriptoras de la ruta se reduce a un punto tridimensional en el subespacio PCA.

La matriz de las proyecciones \mathbf{P} contiene parte de la información de las imágenes originales, pero la dimensión de sus vectores es mucho menor que la dimensión original; esta reducción depende del número de vectores propios utilizados para realizar la transformación. Si $K = N$, la información contenida en \mathbf{P} es la misma que en \mathbf{X}

(puesto que en este caso el error de reconstrucción es nulo [4]), sin embargo, la dimensión de los vectores reducidos es mucho menor: cada proyección, \mathbf{p}_j , tiene sólo K componentes.

3.2 PCA DURANTE LA NAVEGACIÓN AUTÓNOMA

3.2.1 Auto-localización

El proceso es idéntico al descrito anteriormente, la única diferencia está en la información que se utiliza para la navegación. La imagen actual se proyecta en el subespacio PCA calculado en la fase de aprendizaje. Esto producirá un vector K -dimensional en el subespacio PCA. Este vector será comparado con los almacenados en la base de datos relativa a la ruta y se corresponderá con aquél vector que ofrezca un error mínimo en la medida de correlación.

3.2.2 Control

Para cada imagen j de la ruta almacenada, se generan un conjunto de Z ventanas (o subimágenes) de la imagen completa y se utilizan como datos de entrada para realizar otra compresión mediante PCA:

$$\begin{aligned} \mathbf{W}_j &= [\tilde{\mathbf{w}}^1 | \dots | \tilde{\mathbf{w}}^i | \dots | \tilde{\mathbf{w}}^Z] \\ \tilde{\mathbf{w}}^i &= \mathbf{w}^i - \mathbf{m} \\ \mathbf{m}_j &= \frac{1}{Z} \sum_{j=1}^Z \mathbf{w}^i \end{aligned} \quad (8)$$

donde \mathbf{w}^i es cada una de las ventanas y \mathbf{m}_j es la ventana media del conjunto j . Las ventanas se obtienen escaneando la escena original en ambos ejes con un cierto paso.

Para comprimir la información de las ventanas, estas se proyectan en el subespacio PCA:

$$\mathbf{F}_j = \mathbf{U}_j^T \cdot \mathbf{W}_j \quad (9)$$

donde \mathbf{U} es la matriz que define la transformación PCA y contiene los K' vectores propios obtenidos a partir de la matriz de covarianzas de cada conjunto; y \mathbf{F}_j es la matriz de datos reducidos de tamaño $K' \times Z$, donde $K' \leq Z$. Todo este proceso lleva asociado un alto coste computacional, pues se deben obtener los vectores propios de las ventanas de cada imagen de la ruta, sin embargo este hecho no entorpece la navegación puesto que se realiza *off-line* una vez que la ruta ha sido almacenada.

Durante la navegación, se parte de la ventana central en la vista actual y se sigue sobre la imagen de correspondencia. La diferencia está en que ahora no se utiliza la imagen original sino las reducidas mediante la ecuación 9. Para realizar este proceso, una vez que el robot conoce su localización, se proyecta la ventana central de la vista actual en el subespacio PCA. Esta operación devuelve un vector de K' componentes correspondiente a la información más relevante de la ventana (en las direcciones de máxima varianza de los datos). Una vez averiguada su ventana correspondiente, la velocidad de giro puede obtenerse con facilidad.

4 Conclusiones

En este trabajo se han propuesto dos posibles soluciones al problema de la navegación continua de un robot basada en apariencia utilizando únicamente información visual. La primera de ellas, consiste en el uso de una base de datos que contiene imágenes de baja resolución tomadas desde la ruta a seguir. Para llevar a cabo la navegación, se han realizado experimentos con diversos esquemas de control, incluyendo reguladores P, PD y PD con sus parámetros variables. Con estas leyes, el robot es capaz de conocer su posición actual y seguir la ruta en una banda de unos dos metros alrededor de la ruta deseada. Esto puede ser llevado a cabo aunque la escena sufra pequeños cambios (iluminación, posición de algunos objetos, oclusión parcial o fallo en una de las cámaras, etc.). Actualmente, se está trabajando en otros métodos de control, como la lógica difusa. La naturaleza heurística de estos reguladores los hace especialmente útiles en

tareas de navegación, y estos conceptos pueden extenderse directamente a navegación basada en apariencia. La segunda solución propuesta está basada en la reducción de la información a almacenar utilizando subespacios PCA. Este método muestra dos grandes ventajas: el tamaño de los vectores a comparar es mucho menor y se puede calcular la mayoría de la información *off-line*, de modo que toda esta información ya está disponible durante la navegación. Además, el tamaño de los vectores es independiente de la resolución de las imágenes, de modo que se espera que este método trabaje bien en entornos muy desestructurados y con velocidades del robot relativamente altas, debido a la importante reducción del coste computacional.

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'.

Referencias

- [1] Artac, M., Jogan, M., Leonardis, A., (2002) "Mobile robot localization using an incremental eigenspace model", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp 1025-1030.
- [2] Belhumeur, P.N., Hespanha, J.P. and Kriegman, D.J., (1997) "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp 711-720.
- [3] Jones, S.D., Andersen, C., Crowley, J.L., (1997) "Appearance based processes for visual navigation", *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp 551-557.
- [4] Kirby, M., (2001) *Geometric Data Analysis*, Wiley Interscience.
- [5] Kirby, M. and Sirovich, L., (1990) "Application of Karhunen-Loeve procedure for characterization of human faces", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, no. 12, pp 103-108.
- [6] Lebeque, X. and Aggarwal, J.K., (1993) "Significant line segments for an indoor mobile robot", *IEEE Transactions on Robotics and Automation*, vol. 9, no. 6, pp 801-815.

- [7] Lewis, J.P., (1995) "Fast normalized cross-correlation", *Expanded version of paper from Vision Interface*, pp 120-123.
- [8] Maeda, S., Kuno, Y. and Shirai, Y., (1997) "Active navigation vision based on eigenspace analysis", *Proceedings IEEE International Conference on Intelligent Robots and Systems*, vol. 2, pp 1018-1023.
- [9] Matsumoto, Y., Inaba, M., Inoue, H., (1996) "Visual navigation using view-sequenced route representation", *Proceedings of IEEE International conference on Robotics and Automation*, vol. 1, pp 83-88.
- [10] Murase, H. and Nayar, S.K., (1995) "Visual learning and recognition of 3-D objects from appearance", *International Journal of Computer Vision*, no. 14, pp 5-24.
- [11] Nayar, S.K., Nene, N.A., and Murase, H., (1996) "Subspace Methods for Robot Vision", *IEEE Transactions in Robotics and Automation*, vol. 12, no.5, pp 750-758.
- [12] Ohno, T., Ohya, A. and Yuta, S., (1996) "Autonomous navigation for mobile robots referring pre-recorded image sequence.", *Proceedings IEEE International Conference on Intelligent Robots and Systems*, vol. 2, pp 672-679.
- [13] Oja, E., (1983) *Subspace methods of Pattern Recognition*, Research Studies Press. J. Wiley.
- [14] Paya, L., Reinoso, O., Gil, A., Garcia, N. and Vicente, M.A., (2005) "Study of the Navigation Parameters in Appearance-Based Navigation of a Mobile Robot", *Lecture Notes in Computer Science*, vol. 3617, pp 1109-1116.
- [15] Regini, L., Tascini, G. and Zingaretti, P., (2002) "Appearance-based robot navigation", *Proceedings of the Workshop su agenti robotici, Associazione Italiana per l'Intelligenze Artificiale, VIII Convegno*.
- [16] Swain-Oropeza, R., Devy, M., Cadenat, V., (1999) "Controlling the execution of a visual servoing task.", *Journal of Intelligent and Robotic Systems*, vol. 25, no.4, pp 357-369.
- [17] Swets, D.L. and Weng, J.J., (1996) "Using Discriminant Eigenfeatures for Image Retrieval", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp 831-836.
- [18] Turk, M.A. and Pentland, A., (1991) "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp 71-86.
- [19] Ulrich, I. and Nourbakhsh, I., (2000) "Appearance-based place recognition for topological localization", *Proceedings IEEE International Conference on Robotics and Automation*, pp 1023-1029.
- [20] Winters, N. and Santos-Victor, J., (2001) "Information Sampling for optimal image data selection.", *Proceedings of the 9th International Symposium on Intelligent Robotics Systems*, pp 249-257.
- [21] Zhou, C., Wei, T. and Tan, T., (2003) "Mobile robot self-localization based on global visual appearance features", *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pp 1271-1276.