XLIV Jornadas de Automática 2023

https://jautomatica.es

Localización y detección de anomalías utilizando imágenes en un marco bayesiano

Slavic, G.[a,b,*], Marín Plaza, P.[a], Marcenaro, L.[b], Martín Gómez., D.[a], Regazzoni, C.[b]

[a] *Universidad Carlos III de Madrid, Butarque 15, 28911 Leganés (Madrid), España*
[b] *Universidad de Génova, via All'Opera Pia 11, 16145 Génova, Italia*

To cite this article: Slavic, G., Marín Plaza, P., Marcenaro, L., Martín Gómez, D., Regazzoni, C., 2023. Localization and anomaly detection with camera data in a Bayesian framework.
XLIV Jornadas de Automática, 885-890. https://doi.org/10.17979/spudc.9788497498609.885

Resumen

La localización y la detección de anomalías son desafíos importantes en la videovigilancia y en la detección de fallos. Considere el caso de un vehículo que patrulla en una estación de tren: este necesita saber dónde se encuentra e identificar las posibles anomalías, como son, un equipaje abandonado o comportamientos humanos sospechosos. En este artículo, abordamos la localización y la detección de anomalías mediante un marco bayesiano y modelos de aprendizaje profundo. Donde asumimos que el vehículo se mueve en un entorno conocido. Durante la fase de entrenamiento, se construye una Red Bayesiana Dinámica Acoplada a través del agrupamiento de los datos de odometría y el aprendizaje de un Autocodificador Variacional de Kalman utilizando las imágenes de una cámara. Posteriormente, en la fase de prueba, solo se proporcionan datos de video. El enfoque propuesto plantea el uso de un Filtro de Partículas de Salto de Markov Acoplado que aprovecha dicha Red Bayesiana Dinámica para extraer anomalías y localizar el vehículo. Por tanto, las anomalías son tanto una salida del método como un resultado intermedio para guiar el proceso de localización. En este trabajo hemos evaluado el método propuesto en dos conjuntos de datos reales.

Palabras clave: Métodos bayesianos, Filtro de partículas, Robots móviles, Percepción y detección, Filtrado y detección de cambios, Localización

Localization and anomaly detection using images in a Bayesian framework

Abstract

Localization and anomaly detection are important challenges in video surveillance and fault detection. Consider the case of a vehicle patrolling a train station: it needs to know where it is located and identify anomalies such as abandoned luggage or suspicious human behavior. In this article, we tackle localization and anomaly detection using a Bayesian framework and Deep Learning models. We assume that the vehicle is moving in a known environment. During the training phase of the model, a Coupled Dynamic Bayesian Network is built by clustering the odometry data and by learning a Kalman Variational Autoencoder on the camera data. Subsequently, in the testing phase, only video data is provided. A Coupled Markov Jump Particle Filter leverages the Dynamic Bayesian Network to extract anomalies and localize the vehicle. Anomalies are both an output of the method and an intermediate result to guide the localization process. We evaluate the method on two real-world datasets.

Keywords: Bayesian methods, Particle filtering, Mobile robots, Perception and sensing, Filtering and change detection, Localization.

1. Introduction

Los seres humanos construyen con sus sentidos una representación del entorno en que se encuentran. Mediante esta representación, pueden reconocer en qué parte del entorno se encuentran y si algo inesperado está pasando. Como los seres humanos perciben el entorno con sus sentidos, también los vehículos autónomos lo miden con sus sensores, por ejemplo, cámaras, GPS, IMU y LiDAR. Los modelos de Inteligencia Artificial

*Autor para correspondencia: giulia.slavic@edu.unige.it

pueden describir cómo las observaciones de los sensores evolucionan en el tiempo y pueden capturar las correlaciones existentes entre ellos. Así aprender las correlaciones resulta particularmente útil si uno de los sensores deja de funcionar. De esta forma, los modelos de los otros sensores y las correlaciones aprendidas pueden ser usadas para predecir los valores del sensor ausente. Un ejemplo de esta situación se encuentra en el ámbito de la localización visual. Este artículo considera el caso donde los datos visuales y de localización están disponibles en la fase de entrenamiento de los modelos; sin embargo, se supone que solo los datos visuales están disponibles durante la fase en línea. Así el método estima la posición del vehículo mediante las imágenes de la cámara y las correlaciones aprendidas.

Además de la localización, el método propuesto realiza la detección de anomalías. Las anomalías indican la presencia de eventos que no se encontraron cuando el vehículo exploró el entorno por primera vez, y que, consecuentemente, el modelo no aprendió en la fase de entrenamiento. La detección de anomalías y la localización constituyen dos temas cruciales en el campo de la videovigilancia y de la detección de averías (Yang et al. (2021); Avola et al. (2017); Chakravarty et al. (2007)). Por ejemplo, un robot patrulla necesita conocer dónde está localizado y detectar anomalías como un equipaje abandonado o comportamientos humanos sospechosos. De manera similar, un vehículo aéreo no tripulado que supervise un barco en busca de fallos debe conocer su ubicación y ser capaz de detectar anomalías como cables rotos o fugas.

La localización visual, o Visual-Based Localization (VBL), deduce la pose de un agente dentro de un entorno conocido. Los métodos de VBL se pueden dividir en dos grupos principales (Piasco et al. (2018)): *métodos indirectos* y *métodos directos*. Los *métodos indirectos* mapean imágenes del conjunto de entrenamiento en un espacio de características elegido, creando una base de datos de descriptores para imágenes ya vistas. De esta forma, cuando se proporciona una imagen de consulta durante la ejecución del método, el algoritmo busca la imagen en la base de datos con el descriptor más similar. Por otra parte, los *métodos directos* recuperan directamente la pose de la consulta según una referencia conocida. Los métodos de regresión de pose constituyen un subtipo de los métodos directos y se pueden desarrollar mediante Redes Neuronales Convolucionales (RNCs), como por ejemplo PoseNet (Kendall et al. (2015)).

La mayoría de los métodos de VBL usan una sola imagen para deducir la pose. Sin embargo, una secuencia de imágenes aumenta la precisión del método. En consecuencia, algunos artículos han comenzado a aprovechar secuencias de imágenes para estimar la VBL (Lee et al. (2021); Brahmbhatt et al. (2018); Valada et al. (2018); Xue et al. (2019); Li et al. (2019)). En este articulo, suponemos que se nos proporcionan datos secuenciales, y para localizar combinamos métodos de aprendizaje profundo y métodos tradicionales de seguimiento. Además, hipotetizamos que la fase de entrenamiento y la fase de prueba se realicen en el mismo entorno, con condiciones variables. Esta hipótesis es coherente con una aplicación de videovigilancia en la que un robot se mueve siempre en el mismo entorno.

Proponemos combinar una RNC y un modelo basado en Redes Bayesianas Dinámicas (RBDs) (Koller and Friedman (2009); Sucar (2015)), lo que nos permite explicar los resultados obtenidos y evaluar la confiabilidad de la predicción realizada. Por tanto, extendemos la tarea de VBL con el método en Slavic et al. (2021), que aprende dos RBDs sobre los datos de odometría y video para para detectar anomalías. Así se construye un vocabulario compuesto de un agrupamiento de los espacios de estado. Por cada grupo hay una media, una covarianza y un modelo de predicción lineal. Además, se calcula una matriz de transición entre los grupos. El modelo para los datos de video es una variación del Autocodificador Variacional de Kalman (KVAE) propuesto en Fraccaro et al. (2017).

De esta forma, modificamos el método de entrenamiento con los datos de video para combinar los dos modelos y formar un modelo RBD Acoplado (RBD-A). El metodo de Slavic et al. (2021) está enfocado en la detección de anomalías. En cambio, en este articulo, localizamos el vehículo y extraemos anomalías simultáneamente. Para lograrlo, proponemos un Filtro de Partículas de Salto de Markov Acoplado (MJPF-A). El MJPF es un sistema dinámico lineal de conmutación (SLDS) que combina un conjunto de filtros de Kalman (KF) con un filtro de partículas (PF). A través del MJPF-A, se usa la RBD-A aprendida para predecir el valor de los datos de odometría mediante los datos de video y se detectan las anomalías. Las anomalías son tanto un resultado final (que indica si ha ocurrido algo inusual) como un resultado intermedio para guiar la localización. Además pueden explicar posibles fallos del modelo, aumentando la explicabilidad del método.

Las contribuciones del artículo son las siguientes: *i)* el desarrollo de un método basado en las RBDs para acoplar modelos de sensores diferentes y aprender las correlaciones entre ellos; *ii)* el uso del MJPF-A para predecir el valor de una modalidad mediante los valores de otra, y para extraer las anomalías; *iii)* el empleo de las anomalías para guíar el proceso de localización, estimar dónde está fallando y explicar por qué.

2. Descripción del método

El método propuesto se divide en dos partes: *i)* una fase de entrenamiento fuera de línea, en la que se extraen los modelos RBD para odometría e imágenes de video; *ii)* una fase de prueba en línea, en la que solo se proporcionan las imágenes de video y los modelos aprendidos se utilizan para predecir los valores de los datos de odometría y para extraer anomalías. La Figura 1(a) resume la fase de entrenamiento y la Figura 2 la de prueba. La fase de entrenamiento se basa en el método de Slavic et al. (2021), con algunas modificaciones. Está estructurada en dos partes: *i)* el aprendizaje del modelo de odometría, siguiendo Baydoun et al. (2018); *ii)* el aprendizaje del modelo de video, basado en un KVAE (Fraccaro et al. (2017)) modificado. En las secciones 2.1 y 2.2 describimos el entrenamiento del modelo de odometria y de video, respectivamente. El método está explicado en detalle en Baydoun et al. (2018), Slavic et al. (2021) y Fraccaro et al. (2017) donde se puede obtener más información. En 2.3 analizamos la RBD aprendida. Finalmente, en 2.4 exponemos el MJPF-A y la fase en linea.

2.1. Entrenamiento del modelo de odometria

Se proporciona un conjunto de observaciones odométricas $\{x_t^o\}_{t=1...\tau}$, donde τ es el número de instantes de tiempo. Los estados $\{z_t^o\}_{t=1...\tau}$ están correlados con las observaciones a través de una relación lineal:

$$x_t^o = Hz_t^o + \nu_t, \tag{1}$$

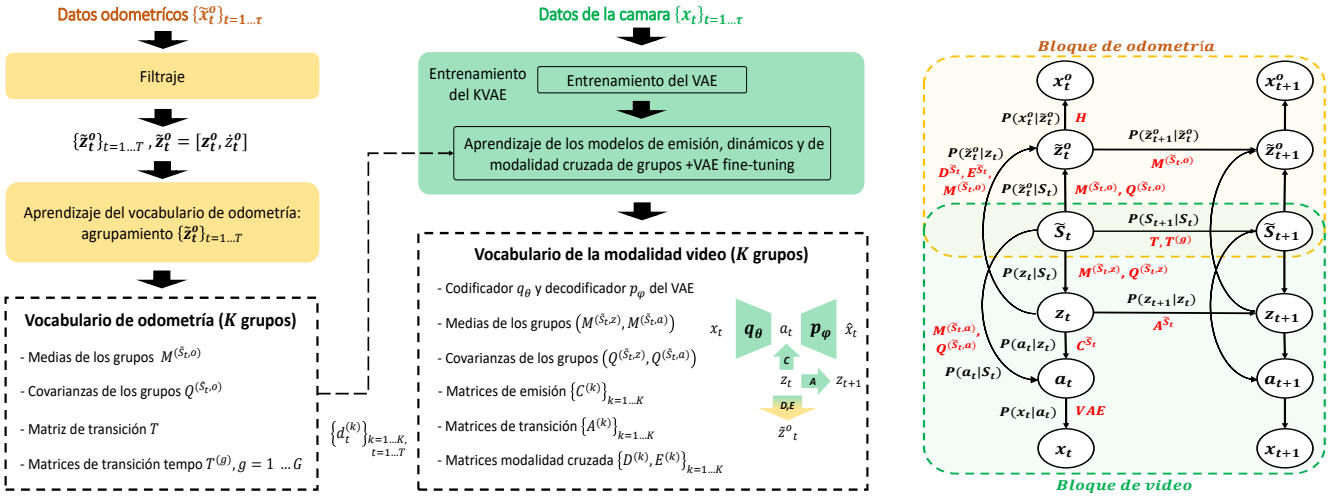


Figura 1: (a): Fase de entrenamiento (secciones 2.1 y 2.2). (b): RBD-A aprendida (sección 2.3).

siendo v_t una distribución gaussiana de media cero con covarianza R^o, y siendo H una matriz identidad.

Los Estados Generalizados (EGs) (Friston et al. (2014)), $\{\tilde{z}_t^o\}_{t=1...\tau}$, contienen los estados y sus derivadas $\tilde{z}_t^o = [z_t^o, \dot{z}_t^o]$. En este artículo, se utilizan las derivadas de primer orden $\dot{z}_t^o$.

Se realiza un agrupamiento de los EGs con el algoritmo Gas Neuronal de Crecimiento (GNG) (Fritzke (1994)). Para cada grupo $\tilde{S}$, con $\tilde{S} = 1 \ldots K$, extraemos: *i)* el centroide $M^{(\tilde{S},o)}$; *ii)* la covarianza $Q^{(\tilde{S},o)}$; *iii)* una matriz de transición T con las probabilidades de pasar de un grupo a otro; *iv)* un conjunto de Matrices de Transición Temporal (MTTs) $T^{(g)}$ con las probabilidades de pasar de un grupo a otro, dado que se han pasado g instantes en el grupo actual, donde $g = 2 \ldots G$, siendo G el tiempo máximo de permanencia en un grupo.

Calculamos la distancia de Mahalanobis $d_t^{\tilde{S}}$ entre cada EG $\tilde{z}_t^o$ y cada grupo. Donde se obtiene un conjunto de $\tau * K$ distancias. A continuación, definimos con $\boldsymbol{d}_t$ el conjunto de K distancias en cada instante t. Estas distancias se proporcionan como entrada al bloque de entrenamiento de video para guiar su fase de aprendizaje.

2.2. *Entrenamiento del modelo de video*

Después de aprender el modelo de odometría, se aprende el modelo de video. Donde se proporcionan un conjunto de observaciones de la cámara $\{x_t\}_{t=1...\tau}$. En cada instante t, se pasa una imagen x_t por el codificador q_θ de un Autocodificador Variacional (VAE) para obtener un estado latente a_t. También se extrae un segundo estado latente z_t, de dimensión menor. a_t captura la información de contenido de las imágenes; z_t se enfoca en la dinámica entre imágenes. Los dos estados latentes están correlados a través de un modelo de pseudo-observación:

$$a_t = \sum_{i=1}^{K} \alpha_t^{(i)} C^{(i)} z_t + v_t \tag{2}$$

El modelo dinámico conecta los valores de z_t en instantes consecutivos:

$$z_{t+1} = \sum_{i=1}^{K} \alpha_t^{(i)} A^{(i)} z_t + \omega_t \tag{3}$$

Las matrices $\{C^{(i)}\}_{i=1...K}$, $\{A^{(i)}\}_{i=1...K}$, representan respectivamente un conjunto de modelos de pseudo-observación y de transición. Se combinan a través de un vector de probabilidades $\boldsymbol{\alpha}_t$, obtenido a partir de las distancias de odometría $\boldsymbol{d}_t$. El elemento i-ésimo del vector, $\alpha_t^{(i)}$, contiene la probabilidad del punto de odometría en t de pertenecer al grupo i-ésimo.

En el método original (Fraccaro et al. (2017)), se utilizan cuatro pérdidas principales para entrenar el modelo: *i-ii)* la pérdida de reconstrucción y el término de divergencia de Kullback Leibler del VAE; *iii-iv)* una pérdida relacionada con los modelos de emisión C y otra con los modelos de transición A. Además, dado que en el caso de la localización también es necesario relacionar un valor del estado latente z_t con su correspondiente odometría EG $\tilde{z}_t^o$, adicionalmente entrenamos un conjunto de matrices $\{D^{(i)}\}_{i=1...K}$ y $\{E^{(i)}\}_{i=1...K}$, tal que:

$$\hat{z}_t^{o,(\tilde{S}_t)} = D^{(\tilde{S}_t)} z_t + E^{(\tilde{S}_t)} + M^{(\tilde{S}_t,o)} + m_t, \tag{4}$$

donde $\hat{z}_t^{o,(\tilde{S}_t)}$ es el EG odometríco estimado en el instante t con las matrices del grupo $\tilde{S}_t$, y m_t es el error residual.

2.3. *Red Bayesiana Dinamica Acoplada aprendida*

Durante el entrenamiento, se aprenden dos RBDs. En la Figura 1(b), las matrices o modelos que representan los enlaces se muestran en rojo.

En la *RBD de odometría*, la matriz de observación H relaciona la observación x_t^o con el EG $\tilde{z}_t^o$. La probabilidad que $\tilde{z}_t^o$ pertenezca a un grupo $\tilde{S}_t$ depende del valor medio $M^{(\tilde{S}_t,o)}$ del grupo y de su covarianza $Q^{(\tilde{S}_t,o)}$. La evolución de $\tilde{z}_t^o$ se predice mediante $M^{(\tilde{S}_t,o)}$. En la *RBD del video*, el VAE actúa como un modelo de observación que relaciona x_t y a_t, mientras las matrices C describen un modelo de pseudo-observación (Eq. 2). Las matrices A constituyen el modelo de predicción sobre z_t (Eq. 3). Al igual que en la RBD odometríca, la probabilidad que z_t pertenezca a un grupo $\tilde{S}_t$ depende del valor medio $M^{(\tilde{S},z)}$ y de la covarianza $Q^{(\tilde{S},z)}$. Un enlace directo entre a_t y $\tilde{S}_t$ también se puede definir a través de $M^{(\tilde{S},a)}$ y $Q^{(\tilde{S},a)}$.

Los dos modelos aprendidos corresponden hasta este punto a aquellos en Baydoun et al. (2018) y en Slavic et al. (2021). En

este artículo, proponemos combinarlos a nivel de grupo, porque comparten el mismo grupo. Consecuentemente, se agrega un vínculo entre el estado z_t del video y el estado odometríco $\tilde{z}^o_t$, representado por la Eq. 4. La predicción a nivel de grupo compartido se realiza a través de la matriz de transición T y las matrices de transición temporal $T^{(g)}$.

2.4. *Filtro de Partículas de Salto de Markov Acoplado*

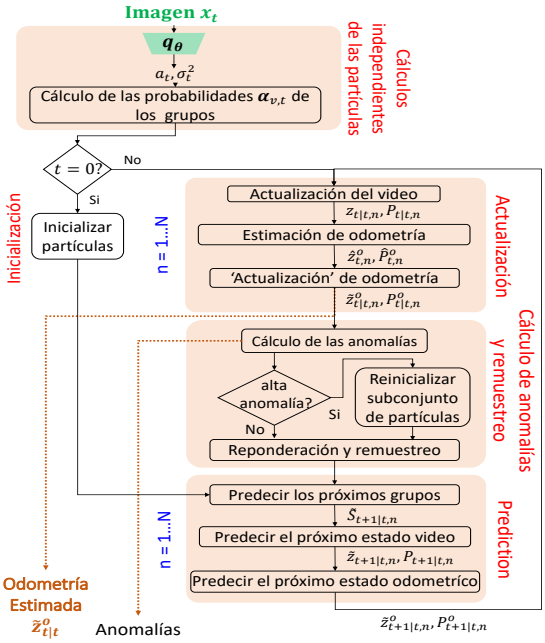


Figura 2: Fase de prueba (sección 2.4).

Durante la fase en línea, se proporcionan solo los datos de video. Derivamos la localización del vehículo y las anomalías, usando la información de la RBD-A en la Figura 1(b). Se construye un MJPF-A. Mientras que el MJPF tradicional en Baydoun et al. (2018) se construyó sobre el modelo de un solo sensor, en este artículo proponemos un MJPF que acopla los modelos de dos sensores, de los cuales solo se observa uno. De esta forma, podemos aprovechar los enlaces aprendidos en la RBD-A para predecir el valor de la odometría a partir del video, al mismo tiempo que realizamos la detección de anomalías.

Un MJPF Baydoun et al. (2018) es un SLDS compuesto por un conjunto de KF a nivel de estado y por un PF a nivel de grupo. Mediante un PF se mantiene un conjunto de N partículas. Cada partícula corresponde a una hipótesis y está asociada a una media, a una covarianza y a un grupo. En cada instante se realiza una fase de predicción y una de actualización. En el MJPF-A propuesto, en cambio, cada partícula está asociada con dos medias (una para el EG de la odometría y otra para el estado de video), dos covarianzas y un grupo.

En los siguientes párrafos, describimos los pasos del algoritmo en cada instante t (Figura 2).

Cálculos independientes de las partículas. En cada instante t se realizan operaciones que no dependen de las N partículas. La imagen x_t se pasa a través del codificador q_θ del VAE para extraer el estado latente compuesto por la media a_t y la varianza σ_t^2. A partir de σ_t^2, la covarianza correspondiente se construye como $\Sigma_t \sim I_L \sigma_t^2$, donde L es la dimensión de a_t. La distancia de Bhattacharya D_B se calcula entre cada estado y grupo de video: $d^{(\tilde{S})}_{v,t} = D_B((a_t, \Sigma_t), (M^{(\tilde{S},a)}, Q^{(\tilde{S},a)}))$. A partir de los K valores $d^{(\tilde{S})}_{v,t}$ se obtiene un vector $\alpha_{v,t}$ con las probabilidades de estar en cada grupo, basado en la observación del video.

Inicialización de las partículas En el primer instante, se inicializan las partículas. A cada partícula se le asocian: un grupo, dos valores medios y dos valores de covarianza (uno para el EG de la odometría y otro para el estado de video). El grupo $\tilde{S}_{t=1,n}$, de la partícula n, se muestrea usando la distribución multinomial descrita por $\alpha_{v,t}$. La media y la covarianza de los estados de video y odometría de la partícula se definen mediante las distribuciones gaussianas $\mathcal{N}(M^{(\tilde{S}_{1,n},z)}, Q^{(\tilde{S}_{1,n},z)})$ y $\mathcal{N}(M^{(\tilde{S}_{1,n},o)}, Q^{(\ tildeS_{1,n},o)})$.

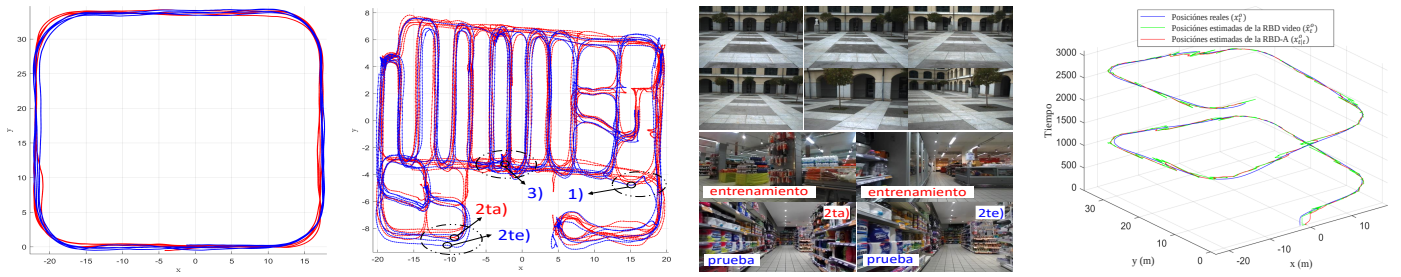
Fase de predicción. En cada instante y para cada partícula, los modelos de predicción aprendidos predicen el siguiente grupo, el siguiente EG de odometría y el siguiente estado de video. A nivel de grupo, la predicción se realiza combinando la matriz de transición T y la MTT $T^{(g)}$, donde g es el tiempo de permanencia en el grupo considerado. Predicimos el estado del video empleando para cada partícula la matriz A del grupo asociado, en un paso de predicción KF estándar. Para la parte de odometría, no es necesaria ninguna modificación con respecto al modelo original propuesto en Baydoun et al. (2018), que usaba un paso KF estándar para predecir que el objeto se moverá con la velocidad media del grupo (es decir, la segunda mitad del vector $M^{(\tilde{S}_{t,n},o)}$).

Fase de actualización. En cada instante, excepto el primero, se realiza la actualización de los estados de video y odometría usando información extraída de la imagen actual. Primero se hace la actualización del estado del video. Para cada partícula se utiliza la matriz C del grupo asociado, es decir, $C^{(\tilde{S}_{t,n})}$. Se realiza un paso de actualización KF estándar, donde la pseudo-observación es el estado a_t, con incertidumbre definida por Σ_t.

Para realizar la actualización del EG de la odometría sería necesaria una observación de la posición del vehículo, pero no está disponible. Por esta razón, durante el entrenamiento del modelo se introdujeron las matrices D y E. Estas matrices obtienen una estimación aproximada del EG de la odometría $\hat{z}^o_{t,n}$ mediante el estado latente $z_{t,n}$ del video (Eq. 4). Hay que notar que, respecto a la ecuación 4, omitimos en $\hat{z}^o_{t,n}$ el símbolo $\tilde{S}$ y usamos n en su lugar, porque cada partícula n está asociada a uno y sólo un grupo.

Se puede realizar una 'actualización' $\tilde{z}^o_{t/t,n}$ de la partícula de odometría como en las ecuaciones estándar del KF, usando $\hat{z}^o_{t,n}$ como observación de odometría. Hay que tener en cuenta que esta no es una actualización real, sino una combinación de dos predicciones, una que usa el modelo de predicción de odometría definido por $(M^{(\tilde{S}_t,o)}, Q^{(\tilde{S}_t,o)})$ y la otra que usa el modelo definido por las matrices $(D^{(\tilde{S}_t)}, E^{(\tilde{S}_t)})$. Empleamos el nombre 'actualizar' porque combinamos las dos predicciones mediante el método estándar de actualización de KF. $\hat{x}^o_{t,n}$ y $x^o_{t/t,n}$, correspondientes a $\hat{z}^o_{t,n}$ y $\tilde{z}^o_{t/t,n}$, se pueden obtener a través del modelo de observación definido por la ecuación. 1.

Cálculo de anomalías y remuestreo. Las anomalías definidas en Baydoun et al. (2018) y Slavic et al. (2021) se extraen después de la fase de actualización. Se realiza una reponderación de las partículas: se otorga mayor peso a las partículas con menor anomalía y pertenecientes a un grupo con mayor probabilidad. Las partículas se vuelven a muestrear cuando el ta-



((a)) ((b)) ((c)) ((d))

Figura 3: (a-b): Trayectorias de entrenamiento (rojo) y prueba (azul) para los conjuntos de datos Icab (a) y Egocart (b). (c): Ejemplos de imágenes de los dos conjuntos de datos. (d): Resultados de la localización sobre parte de los datos de prueba Icab. Los gráficos verde y rojo representan las posiciones estimadas de la RBD de video y de la RBD-A. Las estimaciones $\hat{x}_t^o$ y $x_{t/t}^o$ representan la posición media en cada instante de tiempo de las partículas sobrevivientes después del remuestreo hasta el final de la trayectoria.

maño efectivo (effective size) del filtro está por debajo de un umbral Elfring et al. (2021). Proponemos usar dos umbrales separados. El primero, $N_{th,l}$, se usa al inicio de la trayectoria. Después del primer remuestreo, se sustituye por un segundo umbral $N_{th,h}$, que permanece para el resto del seguimiento. Asimismo, al comienzo de una trayectoria, el algoritmo debe decidir en qué ubicación del entorno se encuentra el agente, y podría necesitar más tiempo para hacerlo correctamente. Por esta razón, establecemos un umbral más bajo al principio, que deja más tiempo al algoritmo antes del remuestreo.

Para evitar divergencias, cuando la j-ésima anomalía abn_j está por encima de un umbral Th_j en un cierto porcentaje P_w de una ventana de tiempo W_h, se reinicializa un porcentaje P_{par} de las partículas. De esta forma, se aprovechan las anomalías para corregir el proceso de localización.

3. Resultados

3.1. *Conjunto de datos usados*

Validamos el método con dos conjuntos de datos:

Icab (Marın-Plaza et al. (2016)). En el conjunto de entrenamiento (6098 imágenes), un vehículo realiza un control perimetral alrededor de un edificio cerrado. En el conjunto de prueba (6.558 imágenes), el vehículo está obstaculizado por la presencia de peatones. Los dos conjuntos están en el mismo escenario. Una dificultad del conjunto de datos para realizar la localización es la simetría del patio en los cuatro lados (ver las primeras cuatro imágenes en la Figura 3(c)).

Egocart (Spera et al. (2018, 2021)). Tanto en el entrenamiento como en el conjunto de prueba (13.360 y 6.171 imágenes), un carro de compras se mueve en un supermercado vacío.

Las Figuras 3(a) y 3(b) muestran las trayectorias de entrenamiento y prueba para los dos conjuntos de datos. La figura 3(c) muestra algunos ejemplos de imágenes.

Los resultados de localización del método propuesto sobre el conjunto de datos de Egocart y el requisito de memoria del modelo entrenado se comparan con los métodos evaluados en Spera et al. (2021). Como la novedad de este documento respecto al método propuesto en Slavic et al. (2021) es la localización y el uso de anomalías para mejorarla, los resultados están relacionados principalmente con estas dos capacidades y se centran menos en la mera detección de anomalías.

3.2. *Discusión de los resultados*

Tabla 1: Error de localización en los dos conjuntos de datos, en metros. EC es “Egocart”, P es “prueba” y E es “entrenamiento”.

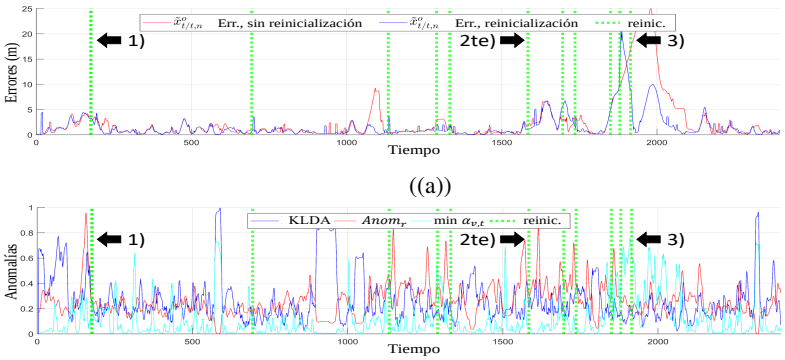
	Método	Err. medio	Err. mediano
Icab (P)	Propuesto, $\hat{x}_{t,n}^o$	1,24	0,77
	Propuesto, final, $x_{t/t,n}^o$	1,12	0,69
EC (E)	Propuesto, $\hat{x}_{t,n}^o$	1,11	0,46
	Propuesto, final, $x_{t/t,n}^o$	0,79	0,32
EC (P)	Propuesto, $\hat{x}_{t,n}^o$	2,34	0,86
	Propuesto, final, $x_{t/t,n}^o$	2,08	0,76
	IR-PNET-VGG16	2,17	1,38
	REG-SVR-PNET-RGB-VGG16	1,96	1,54
	REG-PNET-RGB-POS-IV3	0,42	0,29
	IR-TC-VGG16	0,52	0,28

La tabla 1 muestra los errores de localización obtenidos en los dos conjuntos de datos. Las primeras dos lineas de cada caso se refieren al método propuesto y contienen el error de localización medio y la mediana de las estimaciones de $\hat{x}_{t,n}^o$ (es decir, directamente de la predicción de la RBD de video) y de $x_{t/t,n}^o$ (es decir, el resultado final de la RBD-A). Como el método aprovecha los datos secuenciales, puede localizar el vehículo en un conjunto de datos con simetrías altas como el Icab, con un error medio de 1,11 m. La Figura 3(d) muestra la posición estimada.

Para el caso de Egocart, los resultados están en el rango de los métodos examinados en Spera et al. (2021), es decir, [0,42, 2,17] para el error medio y [0,28, 1,54] para la mediana. La tabla 1 contiene los métodos con rendimientos en los extremos del rango para la media y la mediana.

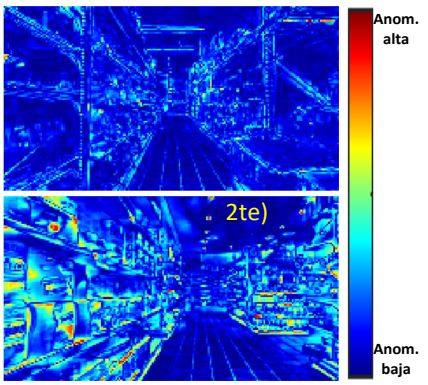
En la Figura 4(a) se muestran los errores finales relacionados con la primera trayectoria de Egocart, en el caso en que no activemos el posible reinicio de un subconjunto de partículas, y en el caso en que lo activemos ($P_{par} = 25\,\%$). El reinicio mejora los resultados de un error medio de $4,40$ m a uno de $1,63$ m. Las barras punteadas verdes muestran los instantes de tiempo en los que se realizan los reinicios.

Fig 4(b) muestra algunas de las anomalías detectadas, explicando la causa del reinicio. Se presentan a continuación algunos ejemplos relacionados con la Figura 3(b). En 1), el reinicio se debe a una anomalía a nivel de grupo (es decir, una secuencia



((a))

((b))



((c))

Figura 4: (a): Errores para la primera trayectoria de Egocart, cuando no se adopta la reinicialización de partículas (rojo) y cuando sí (azul). (b) Ejemplos de anomalías. En ambas figuras, los puntos de reinicio 1), 2te) y 3) son los que se muestran en la Figura 3(b) también. (c): Anomalías en píxeles en un caso normal (arriba) y en uno anormal (abajo) en el conjunto de datos Egocart.

de grupos anomala); en 2te) se debe a un alto error de reconstrucción (ver Figura 4(c)). De hecho podemos notar en la Figura 3(b) y en la Figura 3(c) cómo el punto 2te) está fuera de la zona recorrida en el entrenamiento, siendo el punto 2ta) el más cercano. En 3), los puntos muestran una alta entropía en la probabilidad de pertenecer a los grupos: ver en la Figura 3(b) cómo esta es una zona densa. Debido a esta incertidumbre, algunas partículas se reinicializan.

En resumen, la desventaja del método propuesto respecto a la mayoría de los metodos en Spera et al. (2021) es que una secuencia de imágenes es necesaria; la ventaja es que proporciona una mayor explicabilidad. Consecuentemente, se puede correlacionar la localización con variables de RBD en diferentes niveles de abstracción y extraer anomalías.

4. Conclusiones y trabajo futuro

El artículo propone un método para aprender modelos de un entorno usando dos tipos de sensores, donde se estima el valor de un sensor a partir del otro. El resultado es la localización de un vehículo en un entorno conocido a partir de datos de video y la extracción de anomalías para aumentar la explicabilidad del método. Los datos de video pueden ser sensibles a las condiciones de iluminación del entorno, por tanto, como trabajo futuro, se propone añadir otros sensores (por ejemplo, una unidad de medición inercial) cuando las anomalías indiquen una localización a partir de imágenes de video menos fiables.

Agradecimientos

Este trabajo fue financiado en parte por el Gobierno de España bajo Subvención PID2019-104793RB-C31, PDC2021-1215-17-C31, y PID2021-124335OB-C21, y en parte por la Comunidad de Madrid con la Subvención SEGVAUTO-4.0-CM (P2018/EMT-4362).

Referencias

Avola, D., Foresti, G. L., Martinel, N., Micheloni, C., Pannone, D., Piciarelli, C., 2017. Aerial video surveillance system for small-scale uav environment monitoring. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance. pp. 1–6.

Baydoun, M., Campo, D., Sanguineti, V., Marcenaro, L., Cavallaro, A., Regazzoni, C., 2018. Learning switching models for abnormality detection for autonomous driving. In: International Conference on Information Fusion. pp. 2606–2613.

Brahmbhatt, S., Gu, J., Kim, K., Hays, J., Kautz, J., 2018. Geometry-aware learning of maps for camera localization. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2616–2625.

Chakravarty, P., Zhang, A., Jarvis, R., Kleeman, L., 2007. Anomaly detection and tracking for a patrolling robot. In: Proc. of the Australiasian Conference on Robotics and Automation 2007. pp. 1 – 9.

Elfring, J., Torta, E., van de Molengraft, R., 2021. Particle filters: A hands-on tutorial. Sensors 21.

Fraccaro, M., Kamronn, S., Paquet, U., Winther, O., 2017. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In: Conference on Neural Information Processing Systems. pp. 3601–3610.

Friston, K., Sengupta, B., Auletta, G., 2014. Cognitive dynamics: From attractors to active inference. Proceedings of the IEEE 102 (4), 427–445.

Fritzke, B., 1994. A growing neural gas network learns topologies. In: Conference on Neural Information Processing Systems. pp. 625–632.

Kendall, A., Grimes, M., Cipolla, R., 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. In: IEEE International Conference on Computer Vision. IEEE Computer Society, pp. 2938–2946.

Koller, D., Friedman, N., 2009. Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning. The MIT Press.

Lee, S. J., Kim, D., Hwang, S. S., Lee, D., 2021. Local to global: Efficient visual localization for a monocular camera. In: IEEE Winter Conference on Applications of Computer Vision, WACV. pp. 2230–2239.

Li, C.-T., Siu, W.-C., Lun, D. P., 2019. Semi-supervised deep vision-based localization using temporal correlation between consecutive frames. In: 2019 IEEE International Conference on Image Processing (ICIP). pp. 1985–1989.

Marın-Plaza, P., Beltrán, J., Hussein, A., Musleh, B., Martın, D., de la Escalera, A., Armingol, J. M., 2016. Stereo vision-based local occupancy grid map for autonomous navigation in ros. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 703–708.

Piasco, N., Sidibé, D., Demonceaux, C., Gouet-Brunet, V., 2018. A survey on visual-based localization: On the benefit of heterogeneous data. Pattern Recognit. 74, 90–109.

Slavic, G., Alemaw, A. S., Marcenaro, L., Regazzoni, C., 2021. Learning of linear video prediction models in a multi-modal framework for anomaly detection. In: IEEE International Conference on Image Processing.

Spera, E., Furnari, A., Battiato, S., Farinella, G. M., 2018. Egocentric shopping cart localization. In: International Conference on Pattern Recognition.

Spera, E., Furnari, A., Battiato, S., Farinella, G. M., 2021. Egocart: a benchmark dataset for large-scale indoor image-based localization in retail stores. IEEE Transactions on Circuits and Systems for Video Technology 31, 1253–1267.

Sucar, L. E., 2015. Probabilistic graphical models. Advances in Computer Vision and Pattern Recognition 10.

Valada, A., Radwan, N., Burgard, W., 2018. Deep auxiliary learning for visual localization and odometry. In: IEEE International Conference on Robotics and Automation, ICRA. IEEE, pp. 6939–6946.

Xue, F., Wang, X., Yan, Z., Wang, Q., Wang, J., Zha, H., 2019. Local supports global: Deep camera relocalization with sequence enhancement. In: IEEE/CVF International Conference on Computer Vision. pp. 2841–2850.

Yang, C.-L., Wu, T.-H., Lai, S.-H., 2021. Moving-object-aware anomaly detection in surveillance videos. In: 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance. pp. 1–8.