

# Deep Learning Point Cloud Registration Based on Distance Features

J. Perez–Gonzalez, *Member, IEEE*, F. Luna–Madrigal, and O. Piña–Ramirez

**Abstract**—In this paper, a new method of rigid point cloud registration called Points Registration Learning (PREL) is presented. This algorithm is based on Deep Neural Networks trained by sparse autoencoders and fed with a set of Euclidean and Mahalanobis distance maps. Unlike other reported methods, we do not assume closeness between point clouds or point pairs. This allows registering point clouds with a high degree of displacement or occlusion. PREL algorithm does not require an iterative process, it estimates points distribution non-parametrically and it does not require a finer adjustment using other methods such as Iterative Closest Point (ICP). To evaluate the proposed algorithm, two kinds of point cloud sets were used: one of them corresponds to real scenes acquired with an RGB-D camera and the other set are surface reconstructions. When comparing PREL, ICP and Deep Closest Point (DCP) with Root Mean Square Error (RMSE), using points sets with a high degree of occlusion and displacement, ICP method shows an average RMSE of 98.8, followed by DCP with 32.51 and PREL with 0.75. These results suggest that PREL algorithm can be useful to reconstruct scenes, to scan objects and to register point clouds in any application, given the learning ability of the proposed algorithm.

**Index Terms**—Rigid registration, Point cloud, Deep Learning, Point matching, Object recognition.

## I. INTRODUCCIÓN

EL registro es una técnica utilizada ampliamente para la alineación de imágenes o volúmenes, en otras palabras, un registro es una transformación geométrica que busca alinear una imagen con respecto a otra imagen de referencia [1], [2]. En años recientes, con los nuevos sistemas de captura basados en visión estereoscópica, cámaras de profundidad o sistemas de adquisición 3D de uso clínico como la Tomografía Computarizada (TC) o Resonancia Magnética (RM), se han propuesto gran variedad de algoritmos para el registro de nubes de puntos [3], [4]. Las nubes de puntos se representan como un conjunto de coordenadas que describen la forma de un objeto o escena en un espacio tridimensional principalmente. Sus aplicaciones radican en la detección de objetos, seguimiento de formas, reconstrucción de escenas, análisis y alineación de imágenes médicas usando puntos fiduciales, entre otros [5].

Uno de los principales problemas del registro de nubes de puntos es encontrar la transformación geométrica óptima

Jorge Perez–Gonzalez, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Mérida, Yucatán, 97302, México [jorge.perez@iimas.unam.mx](mailto:jorge.perez@iimas.unam.mx).

Fernando Luna–Madrigal, Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas, Instituto Politécnico Nacional, Ciudad de México, 07340, México.

Omar Piña–Ramirez, Laboratorio de Sistemas Embebidos, Dirección Adjunta de Investigación y Desarrollo, INFOTEC-Conacyt Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, Aguascalientes, México. [omar.pina@infotec.mx](mailto:omar.pina@infotec.mx).

que alinee adecuadamente un par de conjuntos de puntos. Para la reconstrucción o alineación de escenas y objetos, el tipo de transformación geométrica utilizada es rígida; esta transformación produce rotaciones y traslaciones, conservando la rectitud y ángulos entre los puntos, sin efectuar ninguna deformación en el objeto a reconstruir o alinear [6]. De esta forma, el problema de registro rígido de dos conjuntos de puntos se traduce en encontrar los parámetros de traslación y rotación que mejor alineen ambos conjuntos. Sin embargo, el alto grado de occlusión y datos faltantes en las nubes de puntos hacen que el problema sea un desafío.

El objetivo de este trabajo consiste en diseñar un nuevo método de registro rígido de nubes de puntos basado en redes neuronales profundas que aprendan a efectuar la mejor alineación de dos conjuntos de puntos con artefactos de occlusión, datos faltantes o diferente morfología entre las formas a alinear.

### A. Estado del Arte

Dentro del estado del arte, se han reportado diversos métodos de registro basados en correspondencia de puntos como lo es *Iterative Closest Point* (ICP) [7], [8], el cual esta basado en la correspondencia de puntos minimizando iterativamente el error cuadrático medio. Este algoritmo es principalmente usado para alinear curvas o superficies asumiendo la cercanía de pares de puntos. A diferencia de ICP el método *Random Sample Consensus* (RANSAC) [9] produce una estimación de un conjunto de parámetros que describen un modelo matemático basado en un conjunto de datos o nubes de puntos. El objetivo consiste en encontrar iterativamente el modelo que mejor pueda efectuar la correspondencia entre dos conjuntos de puntos. Una ventaja de RANSAC es su capacidad de hacer una estimación robusta de los parámetros del modelo estimándolos con alto grado de precisión, inclusive ante *outliers*. Otros métodos de registro de nubes de puntos son: *Robust Point Matching* [10], el cual aplica un método de alineación determinística convergiendo en una función de energía dada. Chui *et al.* [11] reportaron un algoritmo llamado *Thin Plane Spline Robust Point Matching*, que está enfocado a resolver el problema de registro no rígido ante el problema de *outliers* y ruido. Tsin *et al.* [12] reportaron el algoritmo *Kernel Correlation* (KC) basado en una adecuada selección de un kernel que modela cada nube de puntos a registrar. Papazov *et al.* [13] propusieron un método llamado *Stochastic Global Optimization* el cual atiende el problema del ruido, *outliers* y exactitud de la alineación mediante una minimización global sin condiciones iniciales.

Estos métodos descritos tienen como desventajas, ser sensibles al registro de conjuntos de puntos con alto grado de separación y rotación, debido a que asumen la cercanía entre los puntos a registrar. Además, varios de ellos utilizan al método ICP como etapa final para efectuar un ajuste más fino de los puntos cercanos. Otra desventaja es que algunos métodos como KC asumen funciones de densidad predefinidas y pueden tener limitaciones al modelar una distribución no uniforme del conjunto de puntos. Otra limitante, radica en que varios algoritmos como el ICP o RANSAC maximizan la correspondencia de forma iterativa, lo cual repercute en el tiempo de procesamiento.

### B. Estado del Arte de Métodos de Registro Basados en Aprendizaje Profundo

El aprendizaje profundo o *deep learning* involucra el uso de redes neuronales artificiales de más de tres capas; su principal ventaja es que a partir de su arquitectura de conexión se puede incluir la extracción de rasgos, es decir, no se requiere de un algoritmo adicional para realizarla [14], [15]. Recientemente, se han propuesto algunos algoritmos basados en redes neuronales profundas para procesar nubes de puntos. Tal es el caso de PointNet, la cual es una arquitectura diseñada para la segmentación y clasificación de nubes de puntos [16]. Yew *et al.* [17] propusieron un red llamada 3DFeat-Net, la cual aprende descriptores 3D de nubes de puntos. Las vecindades más cercanas de los *keypoints* encontrados son utilizadas para la estimación de la matriz de transformación rígida mediante RANSAC. Aoki *et al.* [18] modificaron el algoritmo Lucas y Kanade (LK) para generar una red neuronal recurrente e integrarla a la red *PointNet*. Esta modificación permite efectuar el registro de nubes de puntos ante artefactos de oclusión o ruido. Dias *et al.* [19] propusieron una red neuronal profunda para la alineación de nubes de puntos, usando como conjunto de entrenamiento una ecuación que considera de las normas  $L_1$  y  $L_2$  entre un par de puntos dados. Wang *et al.* [20] propusieron una descomposición en valores singulares diferencial junto a una red neuronal profunda para efectuar el registro de nubes de puntos. En ese trabajo propone usar ICP como una etapa final del método, para efectuar un ajuste más fino en la alineación. Estos métodos presentan varias desventajas: son iterativos, lo cual afecta el costo computacional y no aseguran encontrar un mínimo global dado que el número de iteraciones es acotado; algunos de ellos utilizan el método ICP para efectuar un ajuste final, los descriptores de puntos se basan en características salientes o distintivas, esto hace que las formas con pocas regiones distintivas sean difíciles de registrar. Adicionalmente, todos los métodos se basan en la suposición de la cercanía entre conjuntos de pares de puntos a alinear, por lo tanto, muestran un pobre desempeño en conjuntos de puntos totalmente desalineados.

En este trabajo, nosotros presentamos un nuevo método de registro rígido de nubes de puntos basado en una red neuronal profunda alimentada con un conjunto de mapas de distancia euclidiana y Mahalanobis. A diferencia de los métodos reportados previamente, nosotros no asumimos la cercanía entre nubes de puntos o pares de puntos. La distribución de los

conjuntos de puntos es estimada de una forma no-paramétrica, a diferencia de otros métodos que asumen una función de densidad gaussiana. Estas cualidades hacen que el método propuesto pueda alinear conjuntos de puntos con un alto grado de desplazamiento, rotación u oclusión, e incluso conjuntos de puntos con mínima correspondencia. El método es capaz de aprender y estimar adecuadamente los parámetros de rotación y traslación en una sola iteración y no es necesario efectuar un ajuste más fino usando otros métodos como ICP. Estas cualidades son útiles para la reconstrucción de escenas, el escaneo de objetos de cualquier morfología y para el registro de nubes de puntos en cualquier aplicación, dada la capacidad de adaptación del algoritmo propuesto.

## II. METODOLOGÍA

En esta sección se describe el algoritmo propuesto para el registro de nubes de puntos con alto grado de desplazamiento, el cual se basa en una red neuronal profunda entrenada con autoencoders y mapas de distancia. Adicionalmente, se presenta en detalle el proceso de validación del método, así como los datos utilizados en la evaluación.

### A. Algoritmo de Aprendizaje para Registro de Puntos

Dados dos conjuntos de puntos a registrar denotados como  $P_F \in \mathbb{R}^3$  y  $P_D \in \mathbb{R}^3$ ; donde  $P_F$  es un conjunto de puntos fijo,  $P_D$  es una nube de puntos desplazada; ambos conjuntos de puntos son representados en coordenadas cartesianas en 3D. El objetivo consiste en encontrar una matriz de transformación homogénea  $\mathcal{T}$ , tal que efectúe una transformación geométrica

$$P_F = \mathcal{T}\{RP_D + T\}. \quad (1)$$

El problema ahora consiste en encontrar los parámetros de rotación  $R \in \mathbb{R}^3$  y traslación  $T \in \mathbb{R}^3$ , que alineen de forma óptima ambos conjuntos de puntos  $P_F$  y  $P_D$ .

En este trabajo se propone un nuevo algoritmo llamado PREL (*Points Registration Learning*), el cual tiene la cualidad de aprender los parámetros de la matriz de transformación  $\mathcal{T}$ , para efectuar el registro de nubes de puntos de forma óptima. El método está basado en la extracción de mapas de distancia usados para el entrenamiento de una red neuronal profunda. Los detalles del algoritmo propuesto se detallan a continuación.

### B. Mapas de Distancia

Inicialmente se propone caracterizar ambos conjuntos de puntos mediante mapas de distancia euclidiana ( $M_e$ ) y Mahalanobis ( $M_m$ ). La distancia euclidiana es la distancia entre dos puntos de un espacio euclídeo, el cual puede ser  $n$ -dimensional. Por otro lado, la distancia de Mahalanobis es una métrica de similitud multidimensional entre variables aleatorias con la misma distribución de probabilidad [21]. Estas métricas se definen como:

$$M_e = \sqrt{\sum_{i=1}^n (P_{Fi} - P_{Di})^2}, \quad (2)$$

$$M_m = \sqrt{(P_F - P_D)^T \Sigma^{-1} (P_F - P_D)}, \quad (3)$$

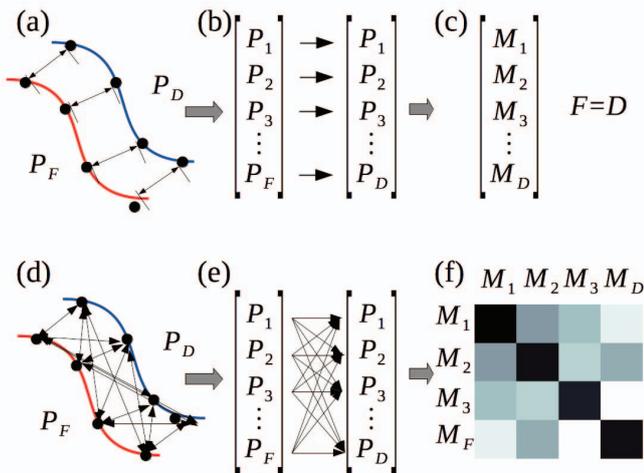


Fig. 1. En la parte superior se ejemplifica del procedimiento empleado por ICP para calcular un vector de distancia. (a) representan los puntos de dos superficies a registrar y la distancia entre los puntos más cercanos, (b) emparejamiento de los puntos más cercanos, (c) vector de distancia resultante. En la parte inferior se ejemplifica el procedimiento propuesto para calcular los vectores o matrices de distancia. (d) puntos de dos superficies a registrar y todas las distancias consideradas, (e) emparejamiento de los puntos y (f) mapa de distancia resultante. En este ejemplo se considera el mismo número de puntos  $F = D$ .

donde  $\Sigma$  representa la matriz de covarianza y  $n$  es la dimensión de las nubes de puntos (para este trabajo se consideran nubes de puntos tridimensionales).

A diferencia otros métodos como ICP, el cual calcula un vector de distancias euclidianas asumiendo la cercanía entre los puntos a registrar, el método propuesto obtiene mapas de distancia considerando todos los puntos. En la parte superior de la Fig. 1, se ejemplifica el procedimiento empleado por ICP para obtener un vector de distancias, Fig. 1(c). En las Fig. 1(a)-(b) se ejemplifica la correspondencia uno a uno entre los puntos a alinear. En contraste, en la parte inferior de la Fig. 1 se ejemplifica como se obtienen los mapas o matrices de distancia, Fig. 1(f), considerando la relación entre cada uno de los puntos a registrar, sin asumir la cercanía o lejanía de estos, Fig. 1(d)-(e). Para este trabajo las distancias entre cada punto fueron calculadas mediante las ecuaciones 2 y 3.

Debido a que el número de puntos  $P_F = \{p_f \mid f = 1, 2, \dots, F\}$  y  $P_D = \{p_d \mid f = 1, 2, \dots, D\}$  se considera diferente  $F \neq D$ . Se propone efectuar un submuestreo aleatorio para igualar el número de puntos de ambos conjuntos  $F' = D'$ .

### C. Entrenamiento de la Red Neuronal Profunda

El entrenamiento de la red neuronal profunda es efectuado mediante *autoencoders* (AE). Los AE tienen una arquitectura de tipo perceptron multi-capa simétrica, es decir, tienen el mismo número de neuronas en la capa de entrada que en la de salida [22], [23]. El objetivo de los AE es construir una función identidad, esto es,  $M = \hat{M}$ ,  $M \in \mathbb{R}^N$  siendo  $M$  y  $\hat{M}$  los mapas de distancia de entrada y salida respectivamente. La Fig. 2 muestra la arquitectura del AE para la etapa de

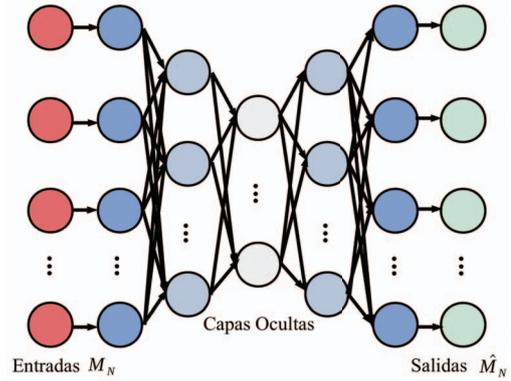


Fig. 2. Arquitectura de la red neuronal profunda en la etapa de entrenamiento usando autoencoders.

entrenamiento. De forma general, la función de decodificación de los mapas de salida estimados  $\hat{M}$  puede denotarse como:

$$\hat{M} = h^{(2)}\{\omega^{(2)}z + b^{(2)}\}, \quad (4)$$

donde  $h^{(2)}$  es la función de transferencia del decodificador (para este trabajo se propone una función sigmoide),  $\omega^{(2)}$  es la matriz de pesos y  $b^{(2)}$  es el vector de sesgo; la segunda capa es representada por el superíndice (2) y  $z$  es una representación codificada de los vectores de entrada de la capa anterior.

Por otro lado, la función de costo utilizada para el entrenamiento esta basada en la suma del error cuadrático medio entre los mapas de entrada  $M$  y los mapas de salida estimados  $\hat{M}$ . Adicionalmente, se pueden añadir funciones de regularización con el objetivo de que cada neurona se dispare únicamente en un número acotado de ejemplos. En otras palabras, cada neurona se especializa en alguna característica que solo esta presente en un subconjunto de ejemplos de entrenamiento [24]. De esta forma, la función de costo se define como:

$$E = \sum_{n=1}^N \sum_{k=1}^K \{M_{kn} - \hat{M}_{kn}\}^2 + \psi \Omega_{L2} + \alpha \Omega_{Disp}, \quad (5)$$

donde  $\psi$  es un factor del termino de regularización  $\Omega_{L2}$ ,  $\alpha$  es el coeficiente de regularización de dispersión  $\Omega_{Disp}$ . Ambos términos de regularización fueron propuestos por Olshausen y Field y están descritos en detalle en [25]. El número de características o atributos que constituye el conjunto de entrenamiento se denota como  $K$  y  $N$  para este trabajo se denomina *nivel de aprendizaje*.

Una vez entrenada la red neuronal, parámetros de rotación y traslación son estimados mediante un proceso de regresión [26]. De esta forma en la etapa de validación la red neuronal tiene como entrada los mapas de distancia y como salida las rotaciones y traslaciones estimadas<sup>1</sup>. Tanto el nivel de aprendizaje (número de nodos de la capa de entrada) y la profundidad de la red son variables que pueden ser asignadas dependiendo la aplicación o el nivel de detalle deseado en el registro. Para evaluar el algoritmo en esta implementación

<sup>1</sup>El código estará disponible bajo petición vía correo electrónico.

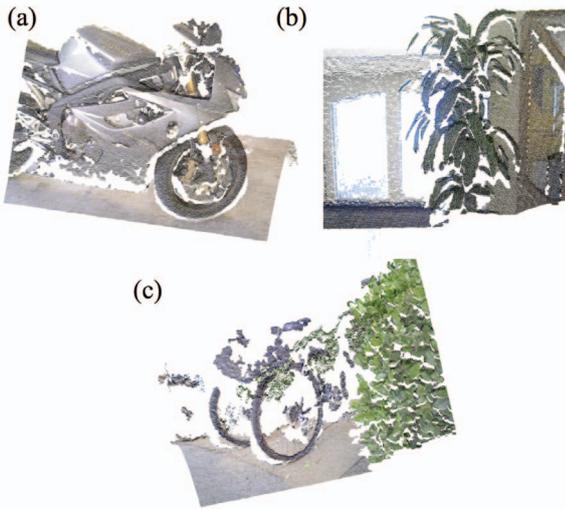


Fig. 3. Nubes de puntos representativas de las escenas utilizadas. (a) motocicleta, (b) planta y (c) bicicleta. Las tres escenas contienen el igual número de puntos ( $NP = 532,512$ ), debido a que se adquirieron con el mismo sistema.

se utilizó un conjunto de 25 capas ocultas, con un nivel de aprendizaje  $N=1000$ . En la siguiente sección se detalla el proceso de evaluación del algoritmo propuesto, así como las características de las nubes de puntos utilizadas y el proceso de evaluación del nivel de aprendizaje.

#### D. Validación

Para la validación del algoritmo propuesto, se utilizaron dos conjuntos de datos de diferente clase los cuales se describen a continuación:

1) *Escenas Reales*: Las escenas utilizadas consisten en imágenes RGB-D adquiridas con la cámara *PrimeSense Carmine*, con una resolución de  $640 \times 480$  pixel a 16-bits; la longitud focal es de 525 mm para ambos ejes. Tanto la imagen RGB como la de profundidad están registradas [27]. Las escenas seleccionadas fueron una motocicleta (Fig. 3(a)), una planta (Fig. 3(b)), y una bicicleta (Fig. 3(c)).

2) *Reconstrucciones de Superficie*: Las reconstrucciones utilizadas consisten en tres modelos de nubes de puntos. Los primeros dos fueron proporcionados por la base de datos de la Universidad de Stanford, y consisten en reconstrucciones de un conejo y un armadillo adquiridos con el escáner *Cyberware 3030 MS* [28], [29]. Estos dos modelos son ampliamente utilizados para la validación de diversos algoritmos de correspondencia de nubes de puntos dada su morfología y variedad de puntos salientes. El tercer modelo utilizado consiste en una esfera 3D construida sintéticamente. Este modelo fue construido con el propósito de verificar la robustez del método; al no tener esquinas o puntos distintivos puede afectar el desempeño de los métodos clásicos basados en correspondencia como ICP. Las tres reconstrucciones se pueden observar en la Fig. 4(a).

Para la validación con datos no vistos del método se aplicaron tres bloques de transformaciones, los cuales incluyen la combinación de diversos grados de rotación, traslación y oclusión, para cada una de las escenas o reconstrucciones

descritas anteriormente. Para la rotación se efectuaron angulaciones aleatorias en rangos de  $45 - 90^\circ$ ,  $90 - 135^\circ$  y  $135 - 180^\circ$  en los tres ejes. En cuanto a la traslación se efectuaron desplazamientos aleatorios de  $20 - 50$  u,  $50 - 80$  u, y de  $80 - 100$  u en los tres ejes. Finalmente, para la oclusión se eliminaron puntos distribuidos de manera no-uniforme (para esto se efectuó una selección de los datos por ordenamiento ascendente o descendente) considerando rangos de  $10 - 20\%$ ,  $20 - 40\%$  y  $40 - 80\%$  del total de los puntos (estas transformaciones se sintetizan en las Tablas I-II). En cada uno de los tres bloques se combinaron los diversos grados de rotación, traslación y oclusión descritos, obteniendo un total de 100 transformaciones para cada nube de puntos.

Debido a que las reconstrucciones y las escenas utilizadas tienen diversas resoluciones, se efectuó una normalización del tamaño de cada nube de puntos en un rango de 0 a 100 unidades adimensionales ( $u$ ), preservando la forma y proporción de cada objeto, esto ayuda a efectuar comparaciones entre las diversas formas. Para medir el desempeño en la alineación se utilizó la Raíz del Error Cuadrático Medio (RMSE), definida como:

$$RMSE = \sqrt{\frac{1}{S} \sum_{j=1}^S (P_{Fj} - P_{Dj})^2}, \quad (6)$$

donde  $P_{Fj} \in \mathbb{R}^3$  representa la nube de puntos de referencia (el conjunto de puntos antes de aplicar las transformaciones descritas) y  $P_{Dj} \in \mathbb{R}^3$  es la nube de puntos desplazada y ocluida. Cada uno de los puntos se denota como  $j$  y  $S$  representa el número total de puntos submuestreado.

El número de datos de entrenamiento de la red neuronal corresponden al nivel de aprendizaje propuesto para este trabajo  $N=1000$ . Para efectuar una evaluación inicial del algoritmo PREL, se efectuó una validación cruzada de 10 vías para cada uno de las formas y escenarios usados en el entrenamiento (un total de 6 validaciones); en cada una de ellas se midió el RMSE promedio y desviación estándar. Dado que, los resultados mostrados en cada validación se consideran despreciables con un promedio global de  $1 \times 10^{-8} \pm 1 \times 10^{-11}$  u, únicamente se reportan los resultados (Tablas I-II) obtenidos de la validación con datos no vistos descritos anteriormente.

Para contrastar los resultados obtenidos del método propuesto, de forma paralela se efectuó la misma prueba con los datos de validación no vistos bajo las mismas condiciones usando ICP y *Deep Closest Point* (DCP) [20]. Para el algoritmo ICP se utilizaron un máximo de veinte mil iteraciones con correspondencia punto a punto. Para DCP se utilizó la red PointNet combinado con ICP para un ajuste más fino. Adicionalmente, se midió el tiempo promedio y su desviación estándar de PREL, ICP y DCP para cada una de las pruebas efectuadas.

Finalmente, se evalúa el desempeño del nivel de aprendizaje en un intervalo  $N=(0,1000)$ , con incrementos uniformes  $N_q = (q-1)50$  para  $q = 1, 2, 3, \dots, 21$ . Para cada  $N$  se calcula su correspondiente error RMSE. De esta forma se puede observar el comportamiento de cada nube de puntos registrada al incrementar el nivel de aprendizaje. Los resultados de

las evaluaciones descritas y su correspondiente discusión se presentan en la siguiente sección.

### III. RESULTADOS Y DISCUSIÓN

Los resultados visuales del método propuesto, el contraste con ICP y DCP se pueden observar en la Fig. 4. En las filas 1–3 se muestran los análisis para cada una de las reconstrucciones analizadas. Las formas originales se muestran en Fig. 4(a); en Fig. 4(b) y 4(c) se pueden observar las nubes de puntos de referencia y desplazadas con su respectivo porcentaje de oclusión. Para este ejemplo se tomaron resultados de los mayores grados de rotación, traslación y oclusión de acuerdo a los experimentos diseñados. Se pueden observar que los niveles de oclusión son elevados, y que existe mínima correspondencia entre los pares de conjuntos de puntos a alinear. En la Fig. 4(d) se muestran los resultados visuales y numéricos respecto al RMSE del método ICP. Se puede notar que en ninguno de los tres casos la alineación fue aceptable. En cuanto a DCP, se puede observar que existen errores notables en la alineación (Fig. 4(e)). Por otro lado, en la Fig. 4(f) se pueden observar los resultados PREL. Cualitativa y cuantitativamente los resultados de la correspondencia entre ambos pares de conjuntos de puntos fue bueno. Se considera que la capacidad de aprendizaje de PREL, y el hecho considerar la relación entre todos los puntos, hace que el método sea robusto ante diversas circunstancias.

En la Tabla I se muestra el análisis detallado del desempeño de PREL, ICP y DCP para cada uno de los experimentos diseñados (considerando un total de 100 transformaciones para cada prueba, las cuales incluyen diversos grados de rotación, desplazamiento y oclusión). Todos los resultados son expresados en unidades adimensionales ( $u$ ) respecto a RMSE. Como se ha mencionado en secciones anteriores, las nubes de puntos fueron normalizadas en un espacio de 0 a 100 unidades preservando su morfología. También se debe recalcar que los datos utilizados en la etapa de validación son datos *no vistos*, lo cual implica que no fueron utilizados para el entrenamiento de PREL ni de DCP.

Se puede observar de forma general en la Tabla I que, los errores aumentaron cuando incrementó el grado de rotación, traslación y oclusión. Con respecto a ICP, el error incrementó drásticamente cuando los pares de nubes de puntos presentaron mayor grado de rotación y traslación entre ellos. Este fenómeno es consistente con las limitaciones reportadas del método ICP. En cuanto a DCP, muestra mejorías con respecto a ICP de forma general, pero aún con menor desempeño que PREL. Esto puede deberse a que DCP es un método que no fue diseñado para nubes de puntos con alto grado de oclusiones. En contraste, PREL mantiene consistencia entre sus resultados sin importar el grado de oclusión, rotación o traslación. Esto puede deberse al aprendizaje del método ante diversos ejemplos, y a que no considera la cercanía de los puntos, si no la relación de distancia entre ellos. De forma global se puede observar un incremento del 463.8% en el error de ICP, de 855.26% para DCP y de 234.3% para PREL, lo cual muestra la robustez del método propuesto teniendo en cuenta que en todos los casos el error fue menor a uno. Considerando

el peor escenario posible (Tabla I columnas 4, 7 y 10), ICP presenta un RMSE de  $98.8u$ , seguido de DCP con  $32.51u$  y PREL con  $0.75u$ . Cabe enfatizar que en todos los casos el algoritmo propuesto supera contundentemente a ICP y DCP como se puede observar en detalle en la Tabla I.

Complementariamente, se midió el desempeño de ICP, DCP y PREL en términos del tiempo de ejecución, lo cual es importante para propósitos de reconstrucción de escenas y registro de diversas formas en tiempo real. En la Tabla II se presenta el tiempo expresado en segundos para cada experimento efectuado. Se puede observar que el tiempo de entrenamiento de PREL disminuye conforme aumenta el grado de oclusión. Esto puede deberse a que, el número de puntos disminuye y por lo tanto el entrenamiento y estimación de los mapas de distancia es más rápido. Por otro lado, se puede apreciar un fenómeno similar para PREL, ICP y DCP de forma global. Se cree, que esto puede deberse a la disminución del número de puntos de forma general. También, se puede observar que, para cada una de las escenas o reconstrucciones analizadas ICP y DCP presenta un mejor desempeño comparado con PREL. De forma global ICP muestra una disminución del tiempo de 7.18 a 5.86 segundos, DCP de 8.02 a 7.36 segundos y PREL de 14.50 a 12.26 segundos. Considerando el peor escenario en cuanto al grado de transformación (Tabla II columnas 4, 7 y 10) ICP presenta un rendimiento promedio de 5.86 segundos, lo cual es menor al mostrado por DCP con 7.36 segundos. En este contexto, PREL presenta el peor desempeño en cuanto a tiempo con 12.26 segundos en promedio. Esto puede deberse a que ICP y DCP solo consideran la correspondencia de pares de puntos y PREL considera las relaciones entre todos los puntos, lo cual afecta el tiempo de ejecución.

Debido a que el desempeño de PREL depende del nivel de aprendizaje  $N$ , se evaluó el error RMSE para cada reconstrucción y escena analizada, considerando el peor escenario (rotaciones de  $135 - 180^\circ$ , traslaciones de  $80 - 100u$  y oclusiones de  $40 - 80\%$ ). En la Fig. 5 se muestran los resultados obtenidos, donde se puede observar que, a medida que aumenta el nivel de aprendizaje, RMSE disminuye en todos los casos. En el caso particular de la esfera se puede notar que error disminuye más rápido que las demás nubes de puntos. Esto puede deberse a que, la esfera presenta una morfología uniforme y no es necesario un nivel de aprendizaje alto para obtener un buen resultado.

Los cálculos y evaluaciones para PREL e ICP se efectuaron en MATLAB2019a; en cuanto al método DCP fue evaluado en Python 3.6, debido a que el código solo está disponible en esa plataforma [20]. Todos los cálculos y evaluaciones fueron realizadas con un procesador Intel® Core™ i5-8500 de 6 núcleos a 4.1 GHz y una memoria DDR4 a 2666 MHz de 8 GiB.

### IV. CONCLUSIONES

En este trabajo se presenta un nuevo método para la alineación de conjuntos de puntos llamado PREL (*Points Registration Learning*). Este algoritmo puede trabajar con conjuntos de puntos que presentan un alto grado de oclusión, rotación y traslación entre ellos. PREL está basado en una

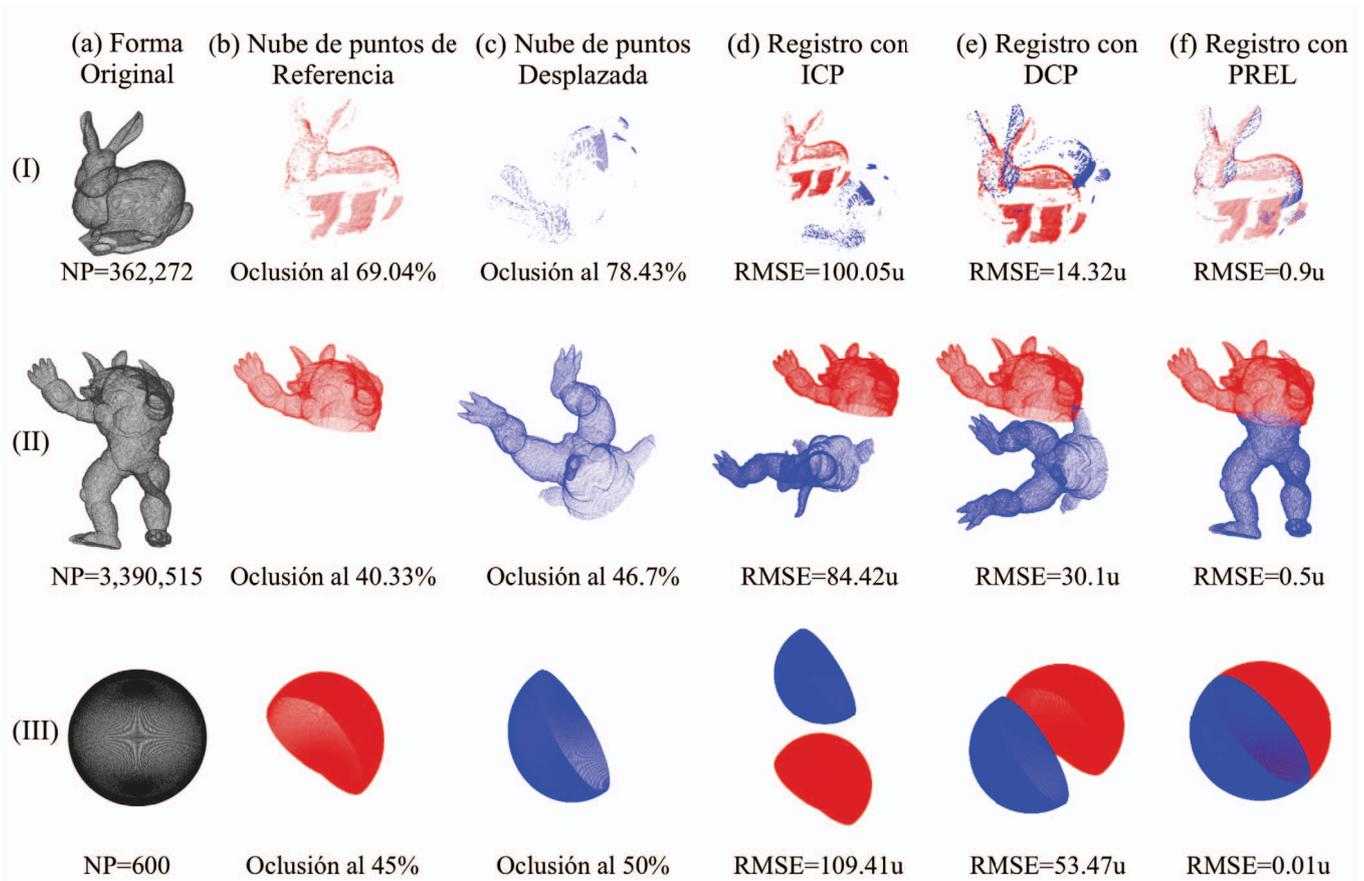


Fig. 4. Resultados representativos de la alineación de nubes de puntos de: (I) la reconstrucción del conejo, (II) armadillo y (III) esfera.  $NP$  representa el número de puntos de cada reconstrucción. (a) Ejemplo de las reconstrucciones originales; (b) nubes de puntos de referencia y su respectivo porcentaje de oclusión; (c) conjuntos de puntos a alinear (en este caso se presentan los resultados con los mayores grados de rotación, traslación y oclusión de acuerdo a los experimentos diseñados); (d) resultado visual del método ICP, (e) resultado usando DCP y (f) resultado representativo del método propuesto (PREL). Los resultados del RMSE fueron obtenidos mediante los datos de validación no vistos.

TABLA I

EVALUACIÓN CUANTITATIVA DEL DESPEÑO DEL MÉTODO PROPUESTO (PREL), ICP Y DCP USANDO RMSE. LOS RESULTADOS SE EXPRESAN EN UNIDADES ADIMENSIONALES  $u$ . PARA CADA EXPERIMENTO SE PRESENTA LA MEDIA Y DESVIACIÓN ESTÁNDAR ( $\mu \pm \sigma$ ) OBTENIDOS DE UN TOTAL DE 100 PRUEBAS CON DATOS DE VALIDACIÓN NO VISTOS; ESTAS PRUEBAS INCLUYEN DIVERSOS GRADOS DE ROTACIÓN, DESPLAZAMIENTO Y OCLUSIÓN. EL PRIMER BLOQUE DE EXPERIMENTOS CORRESPONDEN A LAS NUBES DE PUNTOS DE ESCENAS REALES Y EL SEGUNDO A RECONSTRUCCIONES DE SUPERFICIE

	PREL			ICP			DCP		
	45 – 90°	90 – 135°	135 – 180°	45 – 90°	90 – 135°	135 – 180°	45 – 90°	90 – 135°	135 – 180°
<b>Rotación</b>	45 – 90°	90 – 135°	135 – 180°	45 – 90°	90 – 135°	135 – 180°	45 – 90°	90 – 135°	135 – 180°
<b>Traslación</b>	20 – 50u	50 – 80u	80 – 100u	20 – 50u	50 – 80u	80 – 100u	20 – 50u	50 – 80u	80 – 100u
<b>Oclusión</b>	10 – 20%	20 – 40%	40 – 80%	10 – 20%	20 – 40%	40 – 80%	10 – 20%	20 – 40%	40 – 80%
<b>Motocicleta</b>	0.35±0.2	0.45±0.3	1.24±0.6	23.01±3.4	67.21±15.2	114.34±30.7	3.45±2.5	15.54±4.3	28.49±15.4
<b>Planta</b>	0.45±0.2	0.38±0.2	0.56±0.5	25.85±7.3	58.62±20.8	94.19±41.3	2.96±1.7	13.74±5.2	23.84±12.0
<b>Bicicleta</b>	0.38±0.1	0.41±0.3	0.55±0.4	17.23±3.9	70.10±35.5	89.56±50.8	2.54±1.3	11.99±4.3	25.32±15.3
<b>Conejo</b>	0.21±0.2	0.28±0.3	0.82±0.3	17.88±12.3	70.45±50.3	103.22±61.3	1.76±1.3	12.29±5.7	17.18±7.3
<b>Armadillo</b>	0.23±0.1	0.31±0.2	0.85±0.5	19.36±15.8	62.34±40.3	98.35±67.4	1.46±1.2	9.65±4.9	21.37±10.4
<b>Esfera</b>	0.24±0.2	0.32±0.1	0.38±0.3	21.89±25.3	89.01±70.7	95.11±59.2	10.56±1.3	57.35±20.8	78.86±40.2
<b>Global</b>	0.32±0.8	0.36±0.6	0.75±1.8	21.3±10.1	69.2±42.7	98.8±47.6	3.83±3.3	20.09±18.3	32.51±23.0

red neuronal profunda entrenada con autoencoders, la cual es capaz de estimar de forma óptima un conjunto de parámetros

de rotación y traslación. A diferencia de otros enfoques reportados, en este trabajo no se asume la correspondencia

TABLA II

EVALUACIÓN CUANTITATIVA DEL TIEMPO DE EJECUCIÓN DEL MÉTODO PROPUESTO (PREL), ICP Y DCP. LOS RESULTADOS SE EXPRESAN EN SEGUNDOS. PARA CADA EXPERIMENTO SE PRESENTA LA MEDIA Y DESVIACIÓN ESTÁNDAR ( $\mu \pm \sigma$ ) OBTENIDOS DE UN TOTAL DE 100 PRUEBAS CON DATOS DE VALIDACIÓN NO VISTOS; ESTAS PRUEBAS INCLUYEN DIVERSOS GRADOS DE ROTACIÓN, DESPLAZAMIENTO Y OCLUSIÓN. EL PRIMER BLOQUE DE EXPERIMENTOS CORRESPONDEN A LAS NUBES DE PUNTOS DE ESCENAS REALES Y EL SEGUNDO A RECONSTRUCCIONES DE SUPERFICIE

	PREL			ICP			DCP		
	45 – 90°	90 – 135°	135 – 180°	45 – 90°	90 – 135°	135 – 180°	45 – 90°	90 – 135°	135 – 180°
<b>Rotación</b>	45 – 90°	90 – 135°	135 – 180°	45 – 90°	90 – 135°	135 – 180°	45 – 90°	90 – 135°	135 – 180°
<b>Traslación</b>	20 – 50u	50 – 80u	80 – 100u	20 – 50u	50 – 80u	80 – 100u	20 – 50u	50 – 80u	80 – 100u
<b>Oclusión</b>	10 – 20%	20 – 40%	40 – 80%	10 – 20%	20 – 40%	40 – 80%	10 – 20%	20 – 40%	40 – 80%
<b>Tiempo de entrenamiento</b>	615	611	524	-	-	-	1551	1623	1984
<b>Motocicleta</b>	12.33±2.4	12.04±3.9	11.85±2.7	7.56±2.1	6.88±2.3	6.22±2.5	8.26±2.5	7.34±3.1	7.12±4.2
<b>Planta</b>	14.36±2.7	14.21±3.5	14.33±3.1	7.03±2.0	6.75±2.8	6.42±1.8	7.59±4.2	7.09±4.2	8.10±3.1
<b>Bicicleta</b>	21.87±4.2	18.24±4.8	15.02±3.4	7.84±2.7	6.89±2.8	5.94±1.5	7.96±3.7	7.67±3.8	7.88±4.0
<b>Conejo</b>	13.83±3.7	13.88±3.5	11.56±2.1	7.87±2.8	7.31±3.4	6.54±2.5	8.35±2.9	8.01±3.3	7.57±2.5
<b>Armadillo</b>	13.32±4.1	13.07±4.3	10.17±2.8	7.93±3.1	7.23±2.0	5.76±1.9	8.68±3.1	8.41±3.8	6.98±2.7
<b>Esfera</b>	14.33±4.8	13.89±4.4	12.23±3.2	6.87±2.9	6.55±1.9	6.32±2.1	7.32±2.4	7.22±2.6	6.53±2.1
<b>Global*</b>	14.50±3.8	13.90±3.9	12.26±2.9	7.18±2.6	6.67±2.5	5.86±1.9	8.02±2.5	7.62±2.5	7.36±2.6

\* Sin considerar el tiempo de entrenamiento.

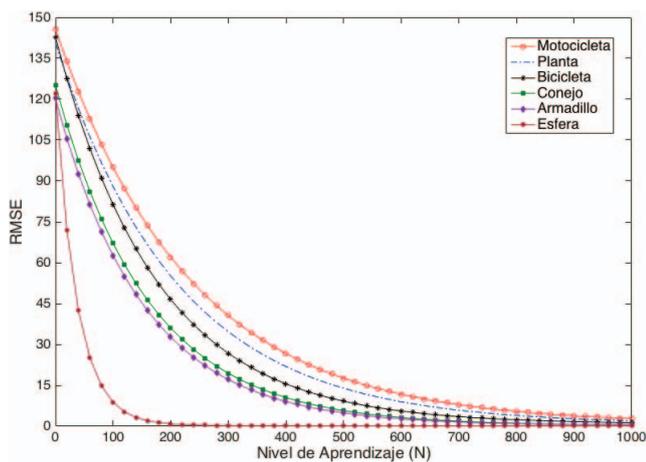


Fig. 5. Evaluación del desempeño de PREL respecto al nivel de aprendizaje  $N$ , con base al RMSE. Cada una de las curvas muestra el desempeño con datos de validación no vistos.

entre pares de puntos, ni la cercanía entre las nubes de puntos. El algoritmo propuesto (PREL) se basa en las relaciones que preservan todos los puntos en su conjunto. Para lograr esto se establecieron mapas de distancia euclidiana y Mahalanobis entre todos los puntos a registrar, lo cual permite encontrar una alineación óptima. Para el aprendizaje de la red se establece un parámetro denominado *nivel de aprendizaje*. El nivel de aprendizaje, consiste en el número de transformaciones que incluyen rotación, traslación y oclusión con las cuales se va a entrenar a la red. Los experimentos realizados, muestran que el nivel de aprendizaje óptimo depende de la morfología de la nube de puntos a registrar. Los resultados encontrados sugieren que formas con pocos puntos salientes o variaciones en su morfología, requieren un grado menor de aprendizaje

comparado con otras formas con mayor grado de variación. Al comparar exhaustivamente PREL, ICP y DCP se encontró que, el algoritmo propuesto muestra mucho mayor precisión que ICP y DCP en casos donde el grado de desplazamiento es alto y alto grado de oclusión. Esto se puede deber a que ICP y DCP se basan en la correspondencia de pares de puntos y PREL estima la relaciones de todos los puntos como un conjunto.

Como desventajas PREL, requiere un entrenamiento previo a efectuar el registro, por lo tanto es necesario generar un conjunto de entrenamiento con un rango estimado del nivel de rotación, traslación y oclusión. Esto es necesario para disminuir el grado de transformaciones en el nivel de aprendizaje. Otra desventaja, es que PREL puede ser sensible a altos grados de ruido o puntos atípicos, por lo tanto como trabajo a futuro se presente incorporar otras características como color o descriptores invariantes de nubes de puntos.

En conclusión PREL muestra ser un algoritmo muy preciso en el registro rígido de nubes de puntos con alto grado de desplazamiento u oclusión. Lo cual puede ser útil en aplicaciones de visión computacional, reconstrucción de escenas o registro de imágenes médicas.

#### AGRADECIMIENTOS

Los autores desean agradecer al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica 2020 (PAPIIT) IA102920, de la Universidad Nacional Autónoma de México por apoyar esta investigación.

#### REFERENCIAS

- [1] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, Oct 2003.

- [2] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable Medical Image Registration: A Survey," *IEEE Trans. Med. Imaging*, vol. 32, no. 7, pp. 1153–1190, Jul 2013.
- [3] T. Pribanić, T. Petković, and M. onlić, "3D registration based on the direction sensor measurements," *Pattern Recognition*, vol. 88, pp. 532 – 546, 2019.
- [4] P. Constante, A. Gordon, O. Chang, E. Pruna, F. Acuna, and I. Escobar, "Artificial Vision Techniques to Optimize Strawberry's Industrial Classification," *IEEE Latin America Transactions*, vol. 14, no. 6, pp. 2576–2581, June 2016.
- [5] A. Narvaez and E. Ramirez, "A Simple 3D Scanner Based On Passive Vision for Geometry Reconstruction," *IEEE Latin America Transactions*, vol. 10, no. 5, pp. 2125–2131, Sep. 2012.
- [6] B. Maiseli, Y. Gu, and H. Gao, "Recent developments and trends in point set registration methods," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 95 – 106, 2017.
- [7] P. Besl and H. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, pp. 239–256, 1992.
- [8] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, Oct 1994.
- [9] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun 1981.
- [10] S. Gold, A. Rangarajan, C. L., S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: pose estimation and correspondence," *Pattern Recognition*, vol. 31, no. 8, pp. 1019 – 1031, 1998.
- [11] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding*, vol. 89, no. 2, pp. 114 – 141, 2003.
- [12] Y. Tsai and T. Kanade, *A Correlation-Based Approach to Robust Point Set Registration*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 558–569.
- [13] C. Papazov and D. Burschka, "Stochastic global optimization for robust point set registration," *Computer Vision and Image Understanding*, vol. 115, no. 12, pp. 1598 – 1609, 2011.
- [14] C. Silva, D. Welfer, F. P. Gioda, and C. Dornelles, "Cattle brand recognition using convolutional neural network and support vector machines," *IEEE Latin America Transactions*, vol. 15, no. 2, pp. 310–316, Feb. 2017. [Online]. Available: <https://doi.org/10.1109/tla.2017.7854627>
- [15] L. M. Q. de Santana, R. M. Santos, L. N. Matos, and H. T. Macedo, "Deep neural networks for acoustic modeling in the presence of noise," *IEEE Latin America Transactions*, vol. 16, no. 3, pp. 918–925, Mar. 2018. [Online]. Available: <https://doi.org/10.1109/tla.2018.8358674>
- [16] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 77–85.
- [17] Z. J. Yew and G. H. Lee, "3DFeat-Net: Weakly Supervised Local 3D Features for Point Cloud Registration," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 630–646.
- [18] H. Goforth, Y. Aoki, A. S. Rangaprasad, and S. Lucey, "PointNetLK: Robust & Efficient Point Cloud Registration using PointNet," in *Proceedings of IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [19] G. Dias Pais, P. Miraldo, S. Ramalingam, V. Madhav Govindu, J. C. Nascimento, and R. Chellappa, "3DRegNet: A Deep Neural Network for 3D Point Registration," *arXiv e-prints*, p. arXiv:1904.01701, Apr 2019.
- [20] Y. Wang and J. M. Solomon, "Deep Closest Point: Learning Representations for Point Cloud Registration," *Computer Vision and Pattern Recognition*, vol. abs/1905.03304, 2019.
- [21] J. M. Cunderlik and D. H. Burn, "Switching the pooling similarity distances: Mahalanobis for Euclidean," *Water Resources Research*, vol. 42, no. 3, 2006.
- [22] C. Xu, L. Wu, and S. Wang, "Unsupervised Dimension Reduction for Image Classification Using Regularized Convolutional Auto-Encoder," in *Advances in Intelligent Systems and Computing*. Springer International Publishing, apr 2019, pp. 99–108.
- [23] O. Aouedi, M. A. B. Tobji, and A. Abraham, "An Ensemble of Deep Auto-Encoders for Healthcare Monitoring," in *Hybrid Intelligent Systems*. Springer International Publishing, mar 2019, pp. 96–105.
- [24] F. Pulgar, F. Charte, A. Rivera, and M. del Jesus, "Choosing the proper autoencoder for feature fusion based on data complexity and classifiers: Analysis, tips and guidelines," *Information Fusion*, vol. 54, pp. 44–60, feb 2020.
- [25] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?" *Vision Research*, vol. 37, no. 23, pp. 3311 – 3325, 1997.
- [26] S. Yu and J. C. Príncipe, "Understanding autoencoders with information theoretic concepts," *Neural Networks*, vol. 117, pp. 104–123, sep 2019.
- [27] S. Choi, Q. Zhou, S. Miller, and V. Koltun, "A large dataset of object scans," *CoRR*, vol. abs/1602.02481, 2016.
- [28] G. Turk and M. Levoy, "Zippered Polygon Meshes from Range Images," in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '94. New York, NY, USA: ACM, 1994, pp. 311–318.
- [29] V. Krishnamurthy and M. Levoy, "Fitting Smooth Surfaces to Dense Polygon Meshes," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 313–324.



**Jorge Perez-Gonzalez** Investigador asociado del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas Unidad Mérida, de la Universidad Nacional Autónoma de México. Concluyó sus estudios de Doctorado y Maestría en Ciencias en Ingeniería Biomédica por la Universidad Autónoma Metropolitana-México. Ha efectuado estancias de investigación en la Universidad Técnica de Múnich-Alemania y colaborado con el Instituto Tecnológico de Massachusetts-Estados Unidos. Sus áreas de interés son el aprendizaje y visión computacional.



**Fernando Luna-Madriral** Ingeniero en Mecatrónica por la Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas del Instituto Politécnico Nacional de México. Ha realizado estancias de investigación en el Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas, de la Universidad Nacional Autónoma de México y en la Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid, España. Sus temas de interés son *machine learning* y la robótica móvil.



**Omar Piña-Ramírez** Coordinador Académico de la Maestra en Sistemas Embebidos e Investigador en INFOTEC-Conacyt. Estudió el doctorado en ciencias con especialización en *machine learning* para el análisis de patrones en la actividad bioeléctrica cerebral con especial énfasis en Interfaces Cerebro-Computadora. Realizó una estancia de investigación en la Universidad Técnica de Graz, Austria en el Laboratorio de Neuroingeniería. Sus temas de interés son el *machine learning*, reconocimiento de patrones y su implementación en sistemas embebidos.