

ALGORITMO PARA LA INTEGRACIÓN/REGISTRO DE NUBES DE PUNTOS TRIDIMENSIONALES BASADO EN TÉCNICAS DE APRENDIZAJE PROFUNDO

Autores:

Juan Ricardo Albarracín Barbosa

Ángel Fabián Gómez Estupiñán

#LaUISqueQueremos

Contenido de la presentación

Introducción

Problema

Objetivos

Solución

- Primera aproximación.
- Segunda aproximación:
 - DPDist: Distancia de Comparación entre nubes de puntos.
 - DPDist: Retropropagación y gradientes.
 - Algoritmo propuesto

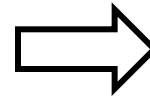
Resultados

Conclusiones

Introducción



LIDAR, cámaras RGBD, etc.



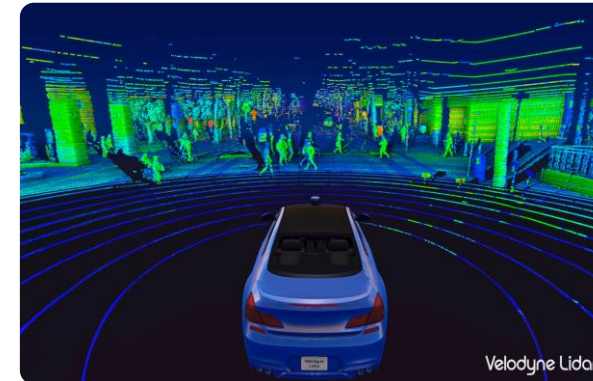
Nubes de puntos

Introducción

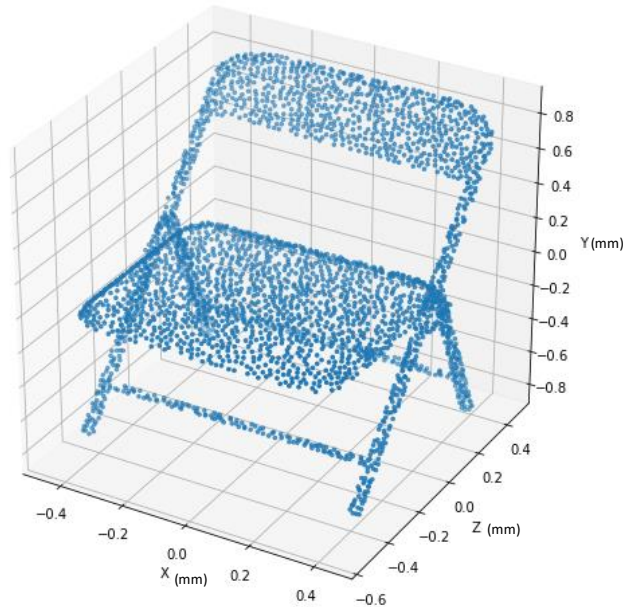
Nubes de Puntos Tridimensionales (3D):

Son los datos tomados para la digitalización 3D, la cuál es la generación de un modelo informático de un objeto o una escena de la vida real.

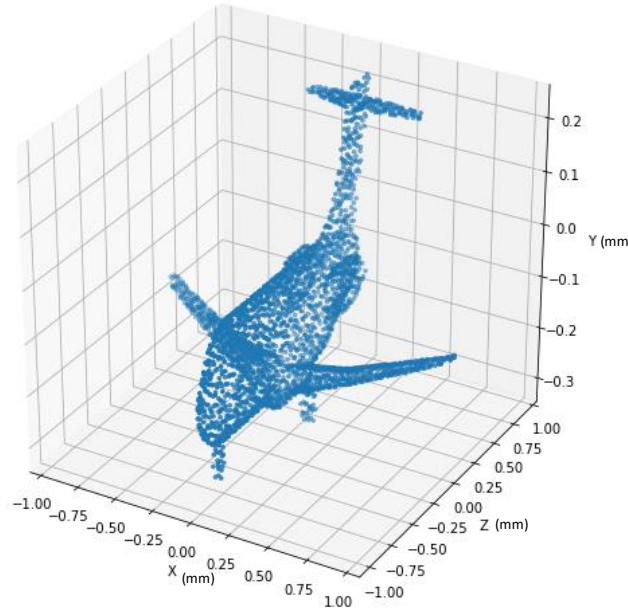
Estos datos proveen una gran cantidad de información espacial y deben ser procesados para determinar los detalles del objeto o escena.



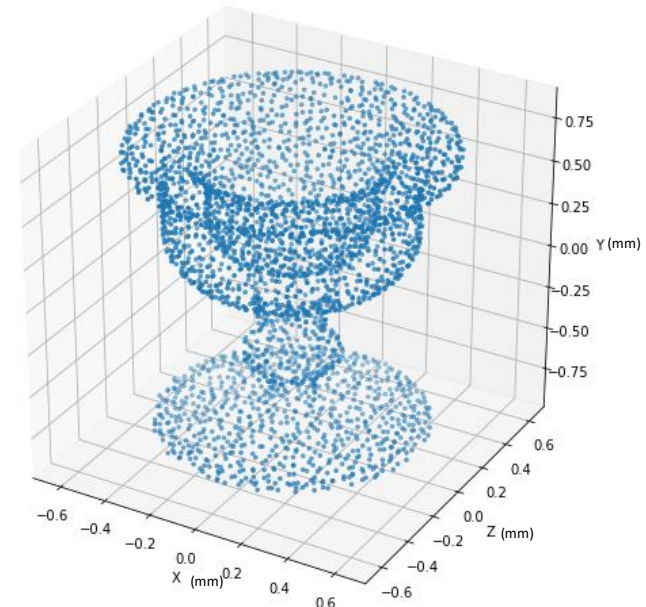
(a) Nube de puntos clase silla



(b) Nube de puntos clase avión



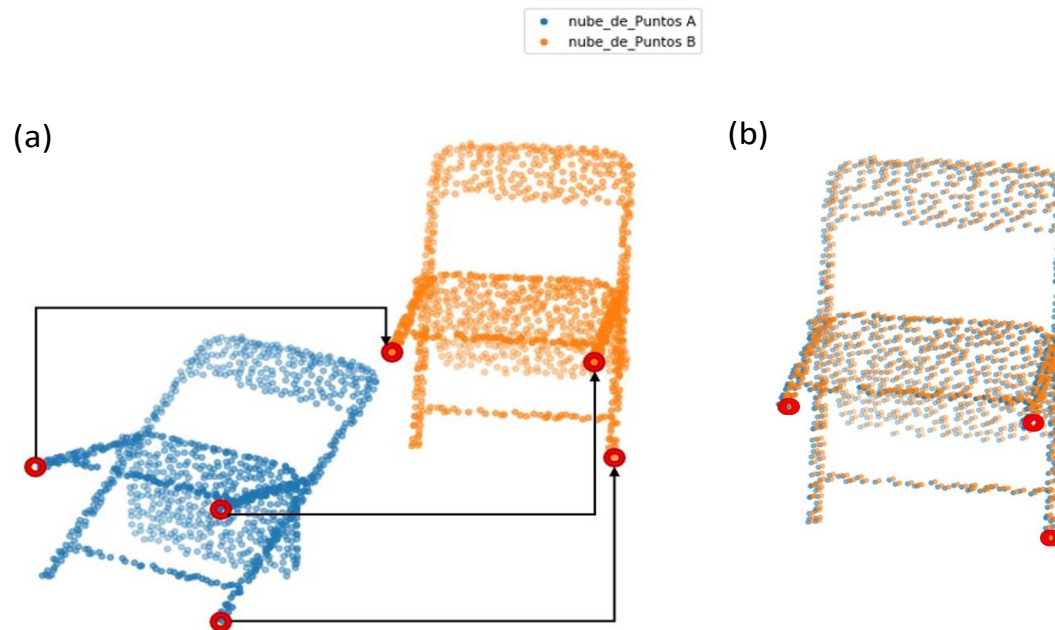
(c) Nube de puntos clase copa



Ejemplos de nubes de puntos de objetos digitalizados de ModelNet40, (a) modelo silla, (b) modelo avión, y (c) modelo copa.

Registro Nubes de Puntos 3D:

Para obtener las nubes de puntos 3D se toman varias muestras, algunas desde diferentes puntos de vista del objeto. El desafío inicia cuando se realiza el proceso de reconstrucción, concretamente es encontrar la transformación rígida T que alinee geoméricamente el par de conjuntos de puntos A y B en un espacio de referencia común.



Matriz de transformación rígida T :

$$A = T \{ R \cdot B + t \},$$

Ejemplo de correspondencia de puntos y registro de dos nubes de puntos disyuntas



Problema

Aún no existe en el estado del arte una herramienta que haga registro de nubes de puntos para reconstrucción 3D de manera óptima y eficiente:

- Procesos costosos para la digitalización de objetos 3D (nubes de puntos).
- Alternativas económicas poco eficientes.
- Necesidad de *automatizar* el proceso de registro de nubes de puntos.
- Algoritmos utilizados para el registro de nubes de puntos reconocidos (ICP, RANSAC, CPD) pero con problemas frente a nubes con datos faltantes (producto de oclusión y/o reflectancia en la fase de obtención).



Objetivos

Objetivo General:

Mediante el uso de técnicas de Aprendizaje Profundo y arquitecturas existentes en el estado del arte, proponer un algoritmo que permita una óptima integración o registro de nubes de puntos tridimensionales en el proceso de digitalización de objetos con alta reflectancia.



Objetivos

Objetivo Específicos:

- Determinar los parámetros más relevantes que influyen en el proceso de selección, correspondencia y alineación de nubes de puntos.
- Estudiar cómo funcionan los algoritmos actuales para realizar procesos de digitalización tridimensional, específicamente de integración de nubes de puntos.



Objetivos

Objetivo Específicos:

- Usar un lenguaje de programación de código abierto (open source) que incorpore las librerías necesarias para modelar el algoritmo que permita la selección y correspondencia de puntos entre nubes.
- Validar, presentar y comparar los resultados obtenidos frente a otros algoritmos que existen en el estado del arte.



Primera aproximación

Registro con una Red Neuronal Profunda (DNN):

En principio, se propuso realizar el registro de nubes de puntos usando una red neuronal profunda alimentada con las coordenadas de nubes de puntos y ángulos de rotación en los ejes x, y, z , basados en el trabajo de Dennis et al., 2021*.



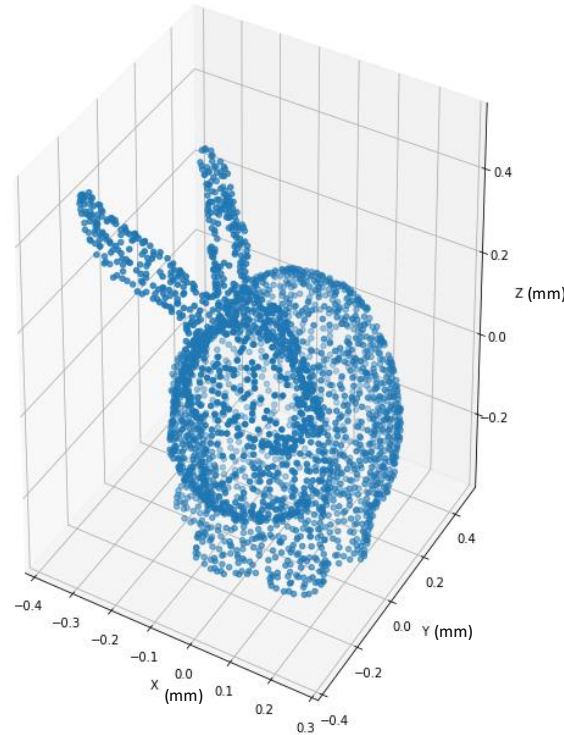
Primera aproximación

Conjunto de datos:

Se usó un conjunto de datos de elaboración propia con 32768 nubes de puntos, cada una con dimensiones de $[2724, 3]$ a partir del modelo del conejo de Stanford*.

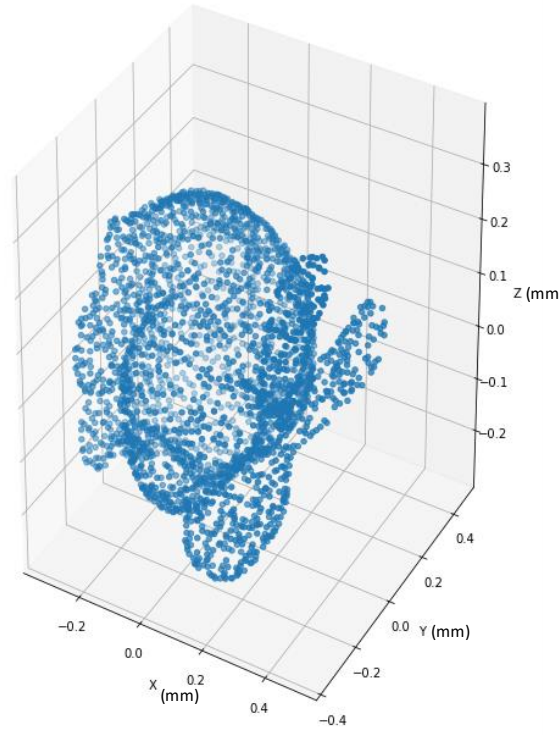
Para realizarlo, se rotó el modelo del conejo en los 3 ejes coordenados cada 0.1 radianes y se registraron dichos valores de rotación para ser el target o etiquetas de los datos que alimentarían a la red.

Nube de puntos original



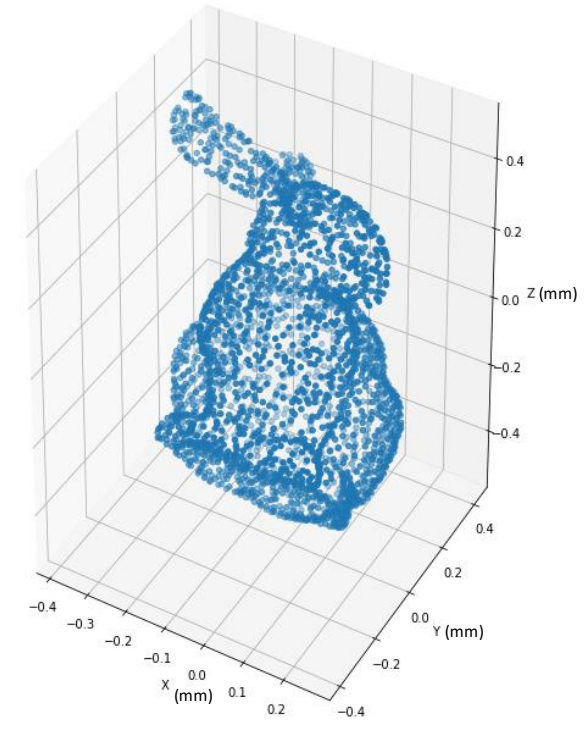
[0, 0, 0]

Nube de puntos rotada 1.6 [rad] en Y



[0, 1.6, 0]

Nube de puntos rotada 3.1 [rad] en todos los ejes



[3.1, 3.1, 3.1]

Tres distintas muestras de datos dentro del dataset, uno es el conejo original y los otros dos son conejos con rotaciones como se especifican en la imagen.



Conjunto de datos:

El total de datos fue dividido en el 80% para entrenamiento y el 20% para test. Así, se entrenó la red neuronal con 26214 modelos (x) y vectores de ángulos (y), y se testearon con 6554 modelos y ángulos.



Primera aproximación

Red Neuronal:

Las nubes de puntos pasan de la capa de entrada con 8172 neuronas a una capa de 1024 neuronas. En la capa de salida se tienen 3 neuronas, una para cada predicción del ángulo x, y, z , respectivamente.

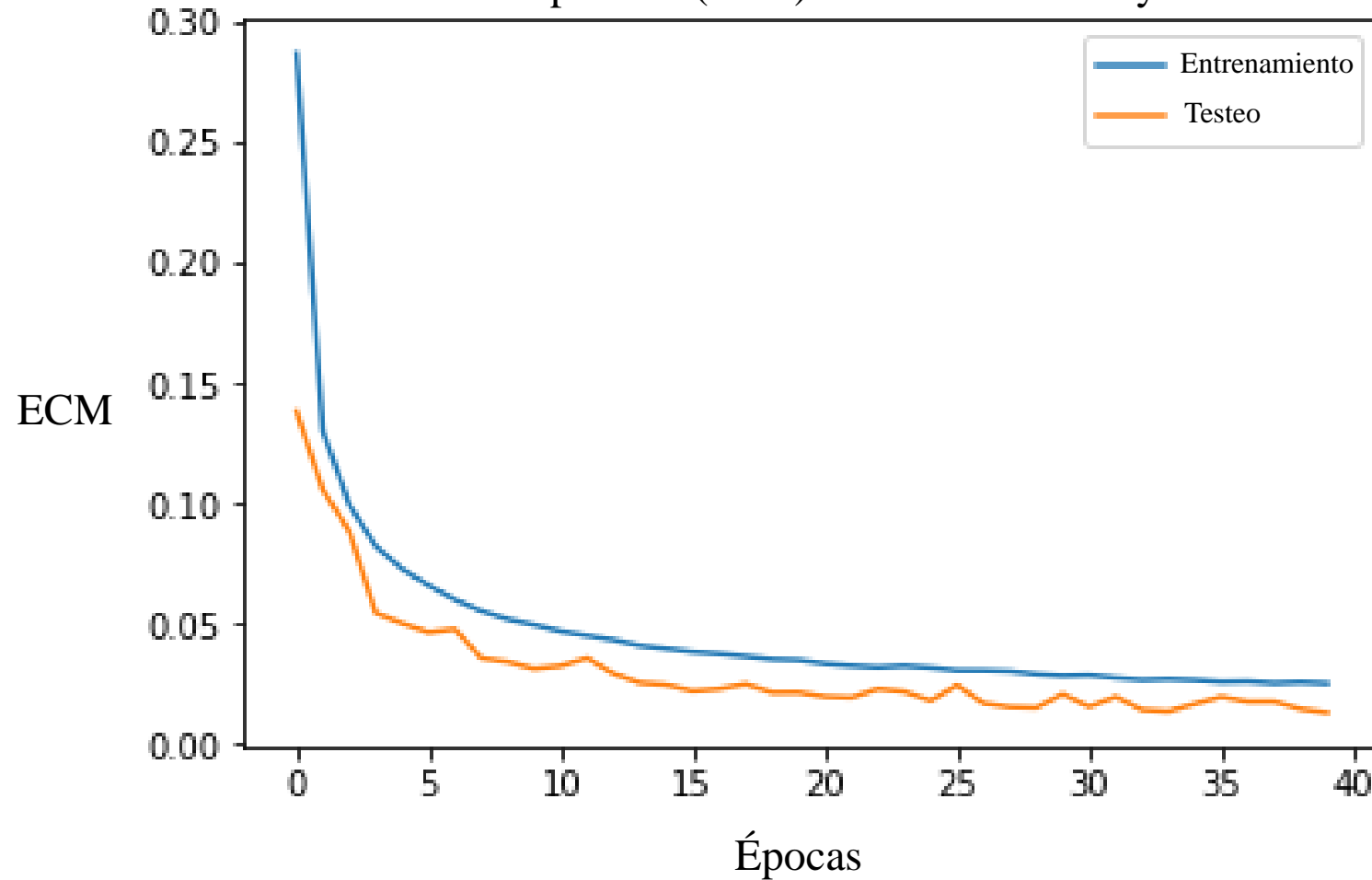
El modelo se entrenó minimizando el error mediante el Gradiente Descendiente Estocástico, con una tasa de aprendizaje de 1×10^{-2} por 40 épocas. Además se usó el Error Cuadrático Medio (ECM) como métrica de costo.



Capas de la red	Dimensión [neuronas]
Aplanada (Flatten) > tanh	8172
Densa (Dense) > tanh	1024
Dropout: 0.5	
Densa (Dense) > tanh	512
Dropout: 0.2	
Densa (Dense) > tanh	256
Densa (Dense)	3

Estructura de la DNN utilizada en la primera aproximación.

Gráfica de pérdida (error) en entrenamiento y test



Resultados en entrenamiento y testeo de la DNN propuesta para la primera aproximación.



Primera aproximación

Pruebas y resultados:

Se realizaron predicciones por la red neuronal para nubes de puntos de la categoría “bunny” diferenciadas por una rotación inicial conocida. La predicción de la red son 3 ángulos de rotación, uno por cada eje coordenado.

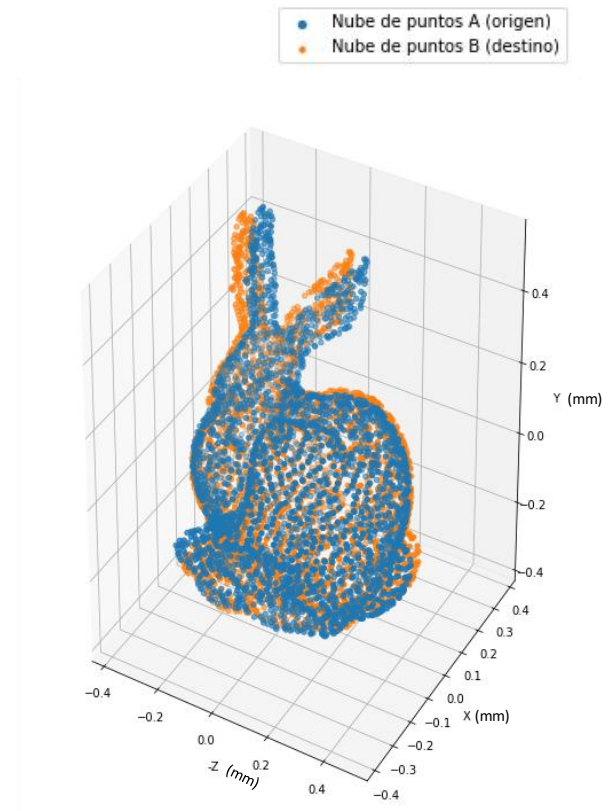
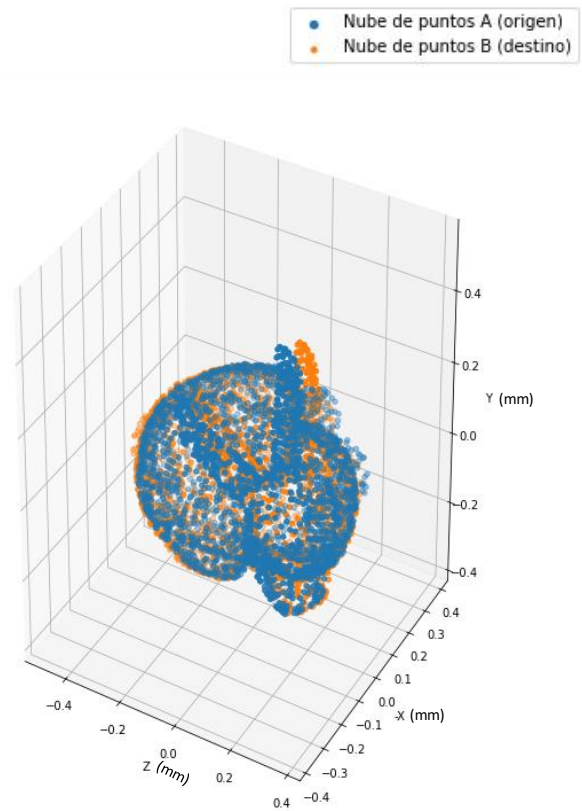
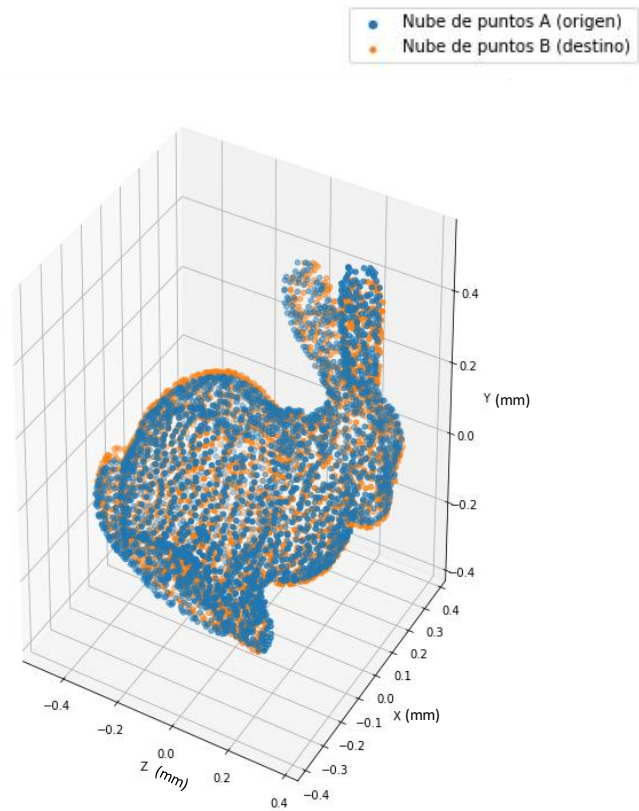
Se presentan los resultados mediante error relativo de los ángulos predichos (valor experimental) con respecto a los iniciales (valor teórico).

$$\text{Error relativo} = \frac{|V_{teo} - V_{exp}|}{V_{teo}} \times 100 .$$



	Eje X	Eje Y	Eje Z
Ángulo inicial [radianes]	1.450	1.509	0.368
Ángulo predicho [radianes]	1.688	1.435	0.668
Error Relativo [%]	16.413	4.903	81.521

Resultados de predicción de la DNN.



Resultado del registro basado en la predicción de la DNN.



Primera aproximación

Conclusiones:

El modelo solo aprenderá a registrar nubes de puntos del modelo “bunny” (Conejo de Stanford), u otras nubes cuya estructura, tamaño y dimensiones sean muy similares o iguales al modelo con el que se realizó el entrenamiento de la red.

Si se quisiera registrar otra nube de puntos de otra clase, no se obtendrían óptimos resultados.



Primera aproximación

Conclusiones:

Para solucionar esto, se tendría que entrenar la red con la máxima variabilidad de clases posible, y, aun así, esto no garantizaría buenos resultados debido a que la red sigue procesando los datos en términos de coordenadas.

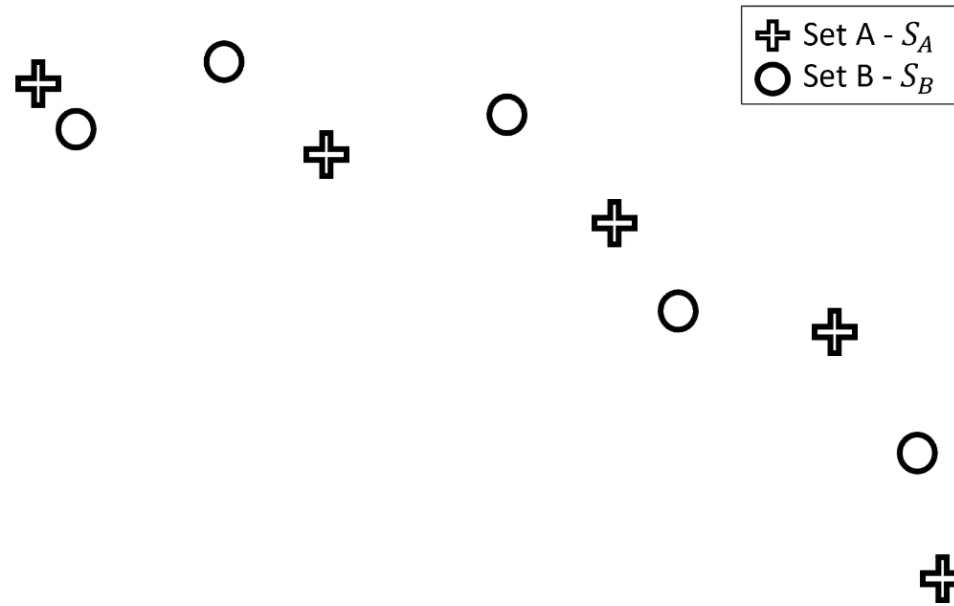
Se establece la necesidad de encontrar una métrica de comparación entre dos nubes de puntos para generalizar el problema de registro de una clase a cualquier nube de puntos.



Segunda aproximación

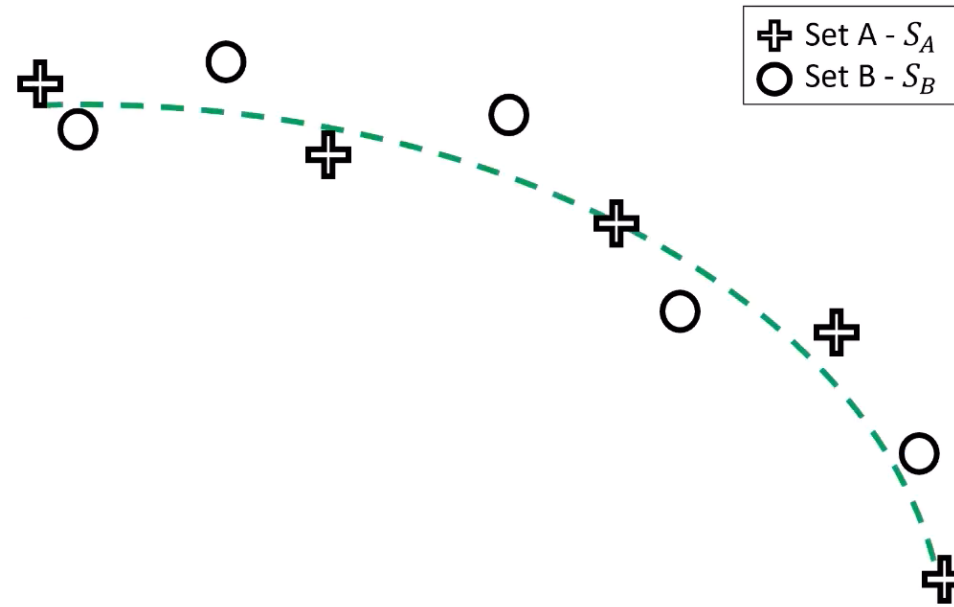
DPDist: Comparación (3DmFV, Superficie subyacente, y métrica de comparación):

El método de Distancia Profunda de Nubes de Puntos, conocido por su nombre en inglés Deep Point Cloud Distance (DPDist)*, se usa para comparar dos nubes de puntos a partir de la estimación de una métrica cuantitativa entre ellas. Esta métrica se define en términos de “distancia” de los puntos desde una nube de puntos hasta la **superficie continua subyacente** correspondiente a la otra nube.



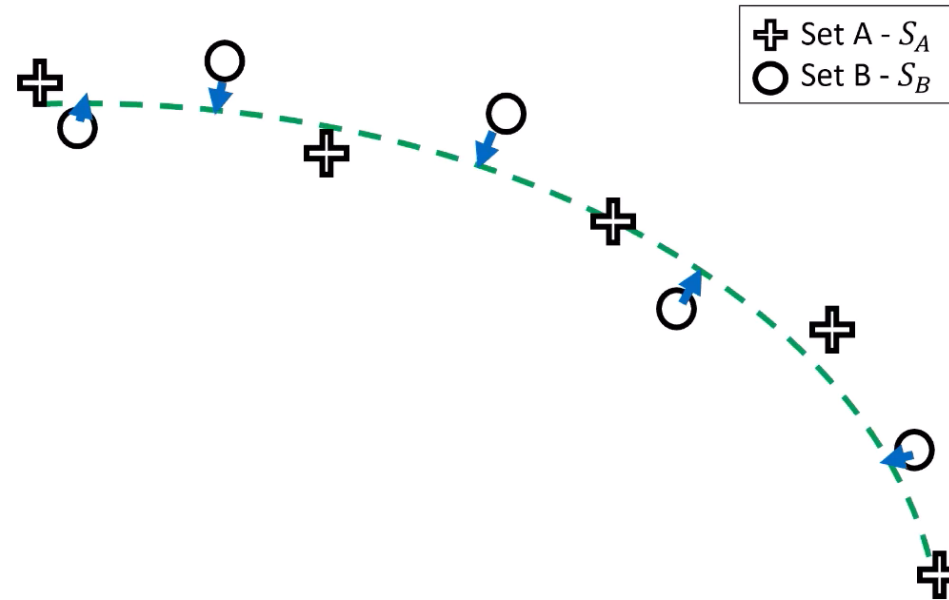
Distancia de cada punto a la **superficie subyacente estimada**

$$D_{DPDist}(S_A, S_B) = \frac{1}{|S_A|} \sum_{a \in S_A} \hat{D}(a, B) + \frac{1}{|S_B|} \sum_{b \in S_B} \hat{D}(b, A)$$



Distancia de cada punto a la **superficie subyacente estimada**

$$D_{DPDist}(S_A, S_B) = \frac{1}{|S_A|} \sum_{a \in S_A} \hat{D}(a, B) + \frac{1}{|S_B|} \sum_{b \in S_B} \hat{D}(b, A)$$



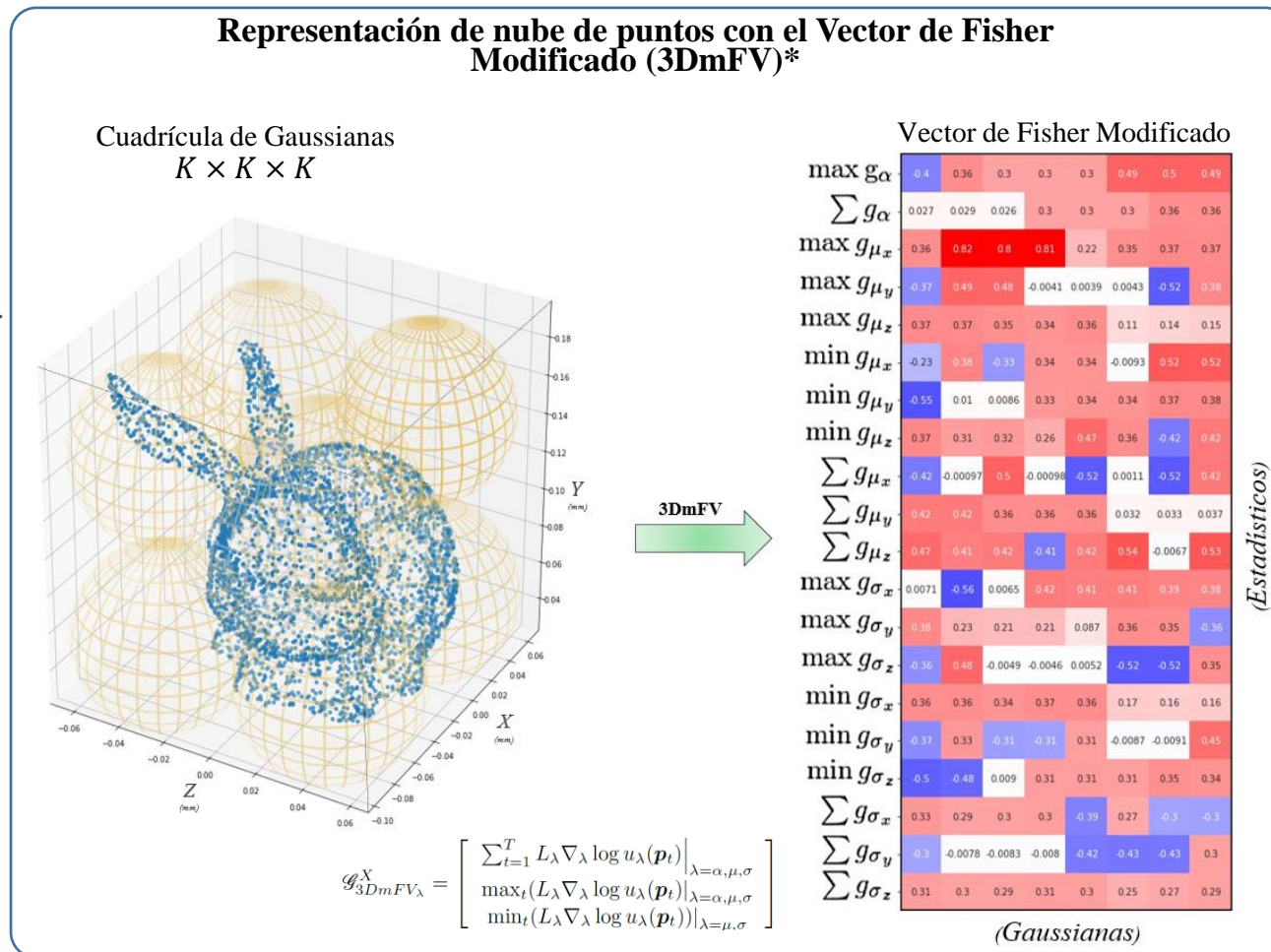
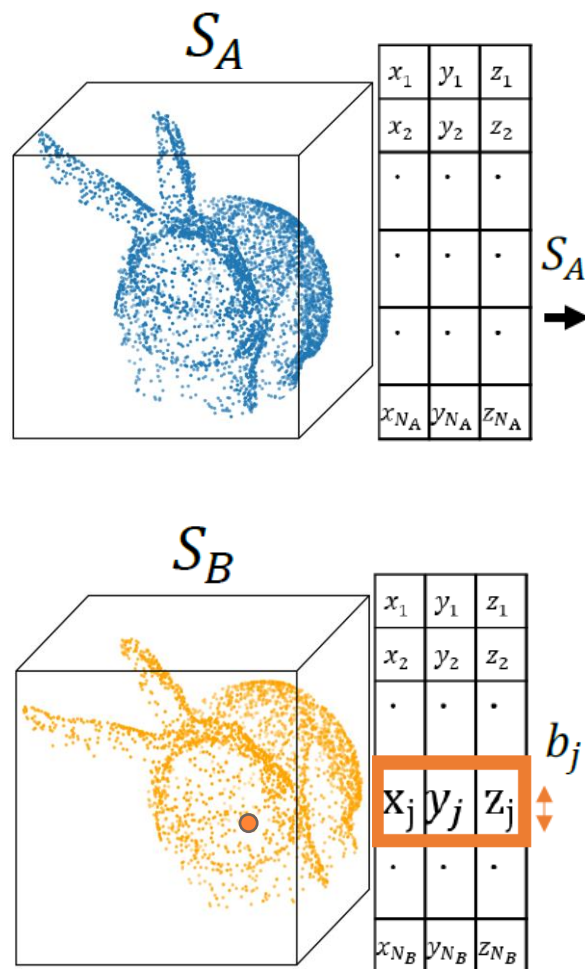
Distancia de cada punto a la **superficie subyacente estimada**

$$D_{DPDist}(S_A, S_B) = \frac{1}{|S_A|} \sum_{a \in S_A} \hat{D}(a, B) + \frac{1}{|S_B|} \sum_{b \in S_B} \hat{D}(b, A)$$

Funcionamiento del modelo DPDist



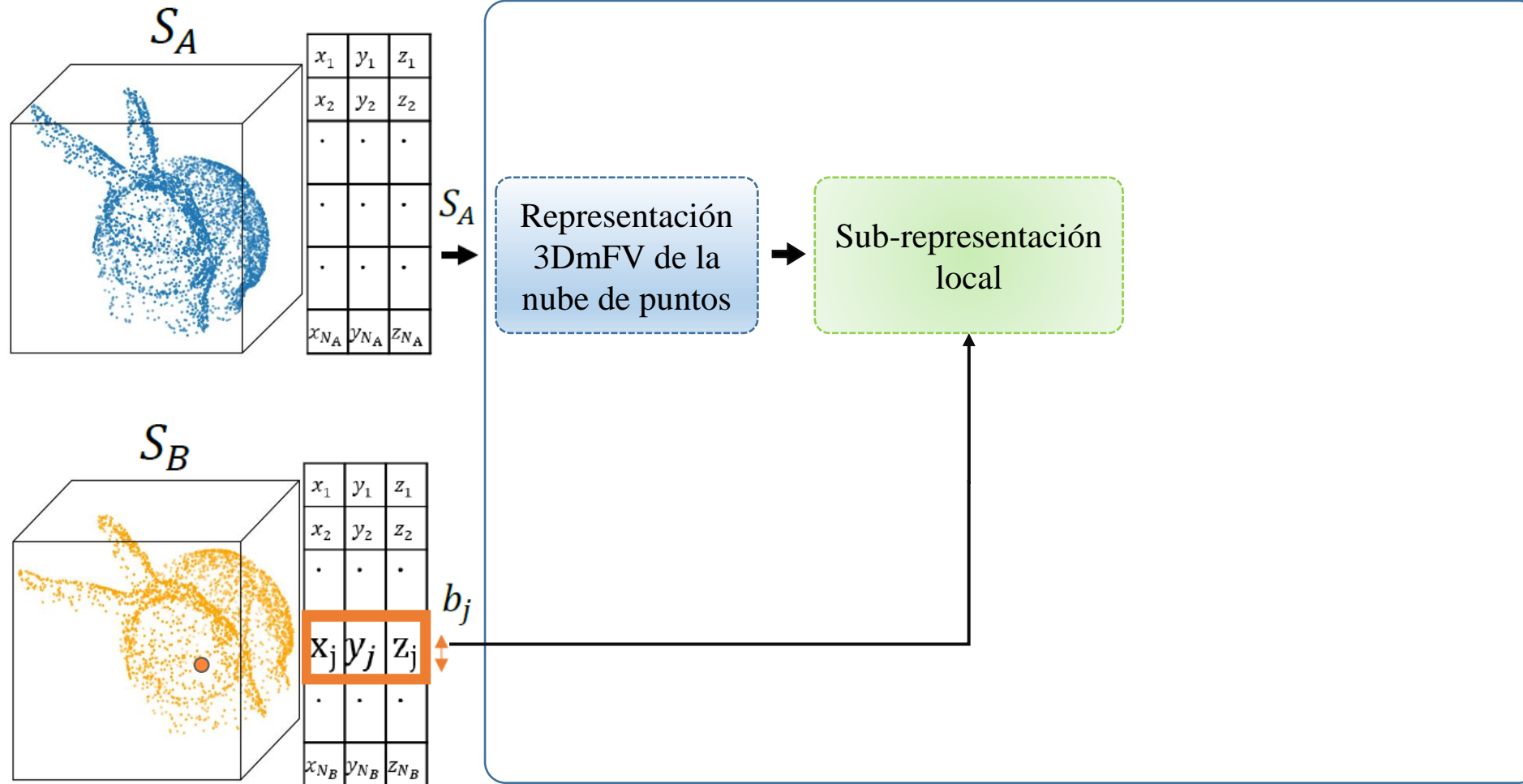
Universidad
Industrial de
Santander



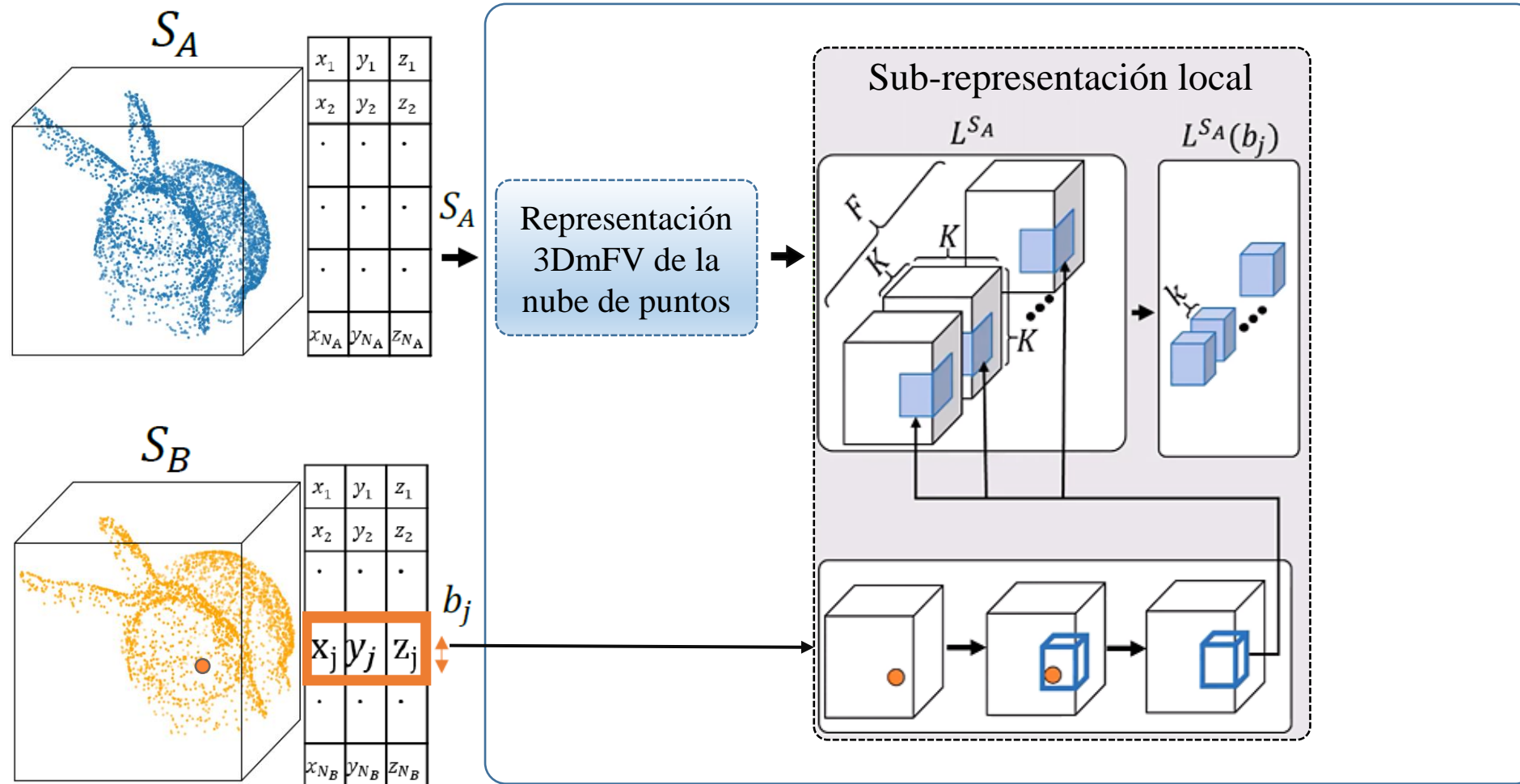
Funcionamiento del modelo DPDist



Universidad
Industrial de
Santander



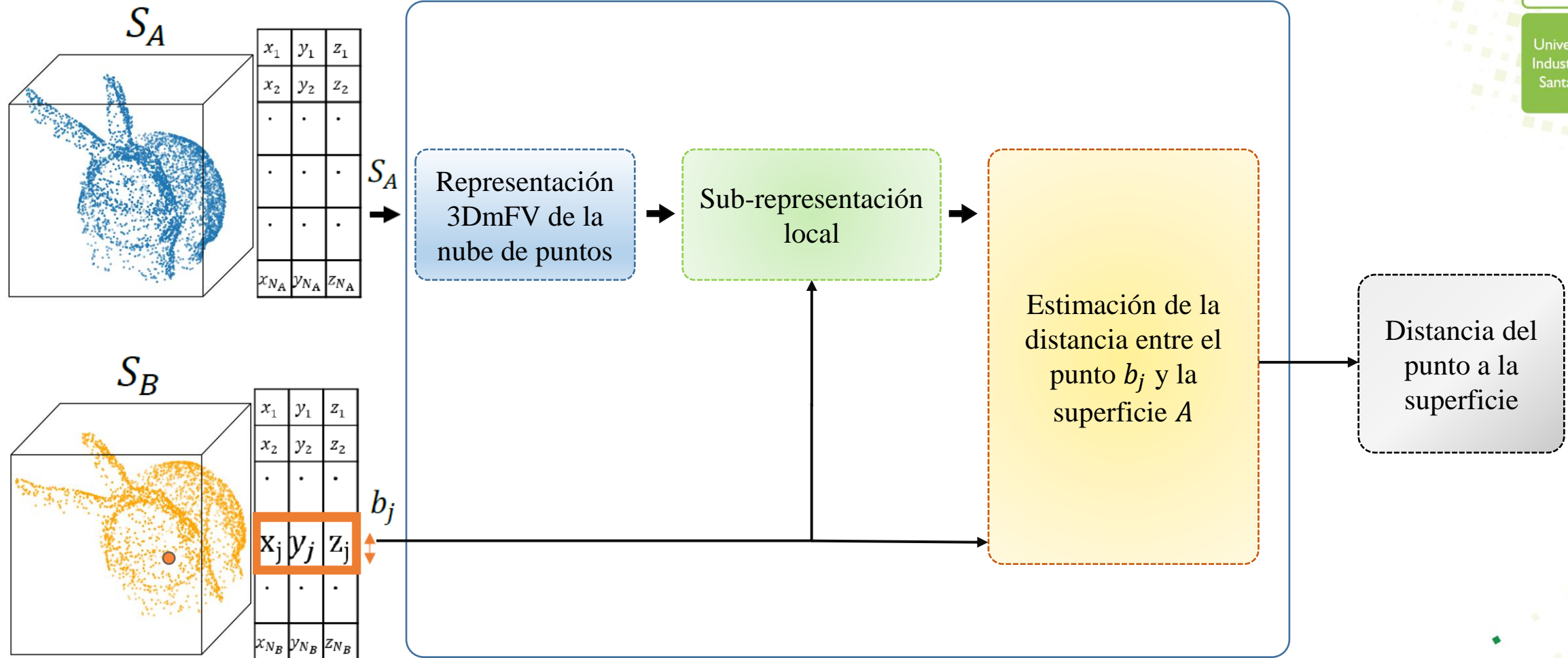
Funcionamiento del modelo DPDist



Funcionamiento del modelo DPDist



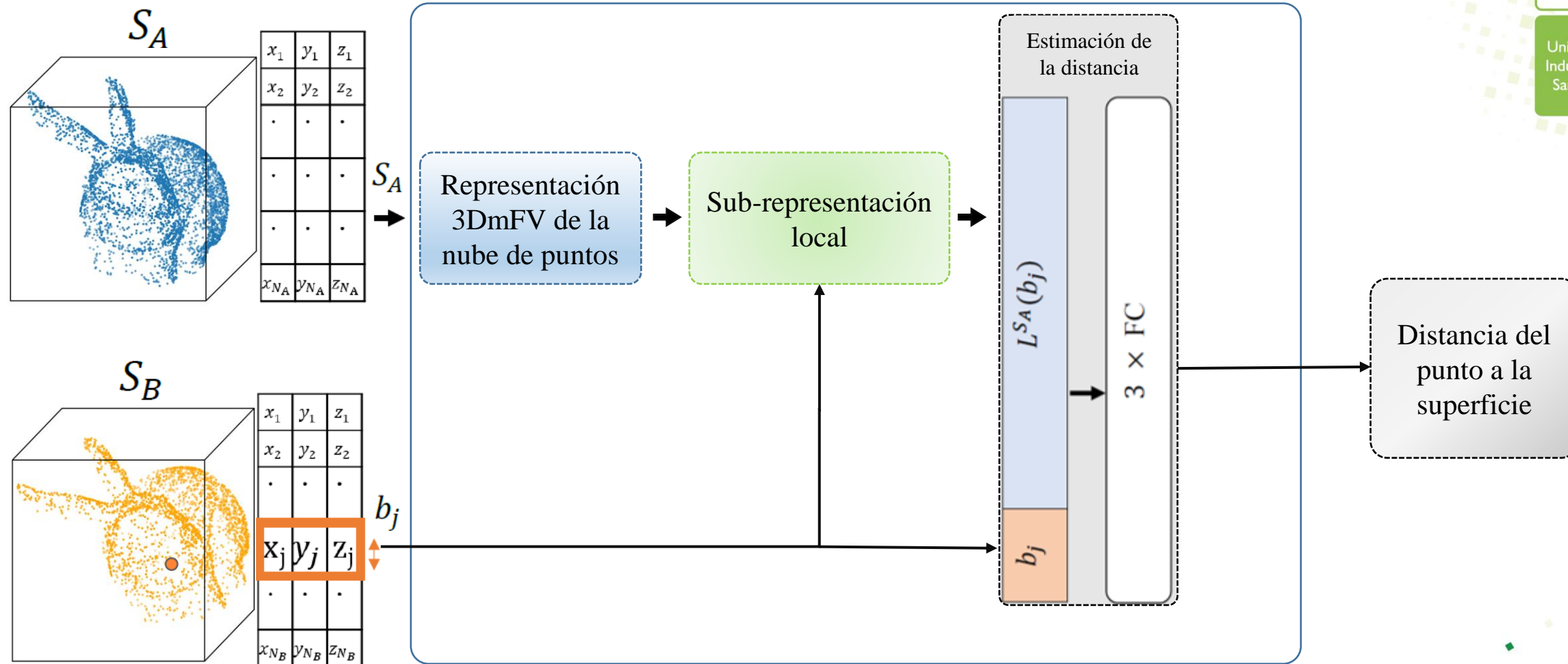
Universidad
Industrial de
Santander



Funcionamiento del modelo DPDist



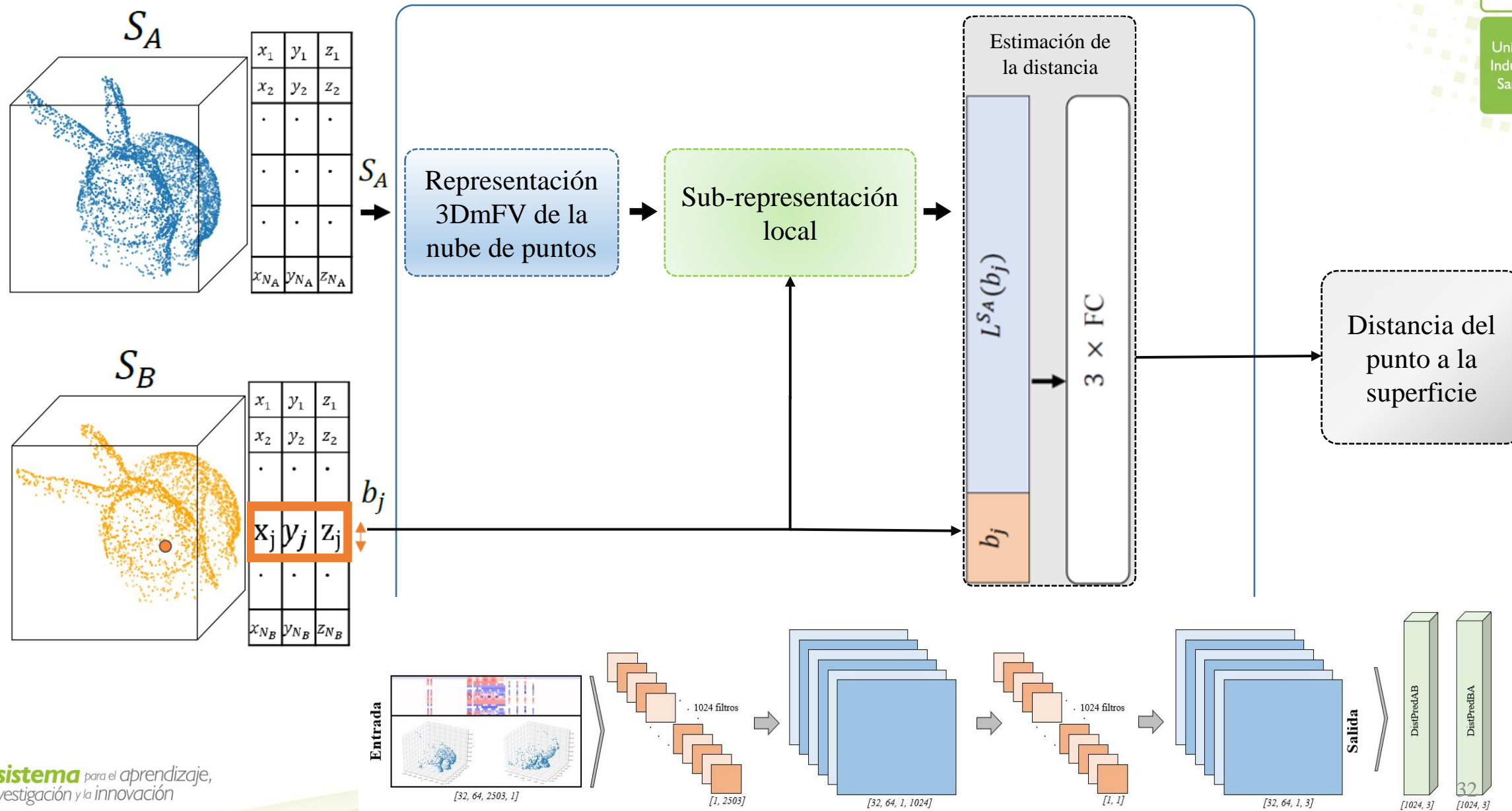
Universidad
Industrial de
Santander



Funcionamiento del modelo DPDist



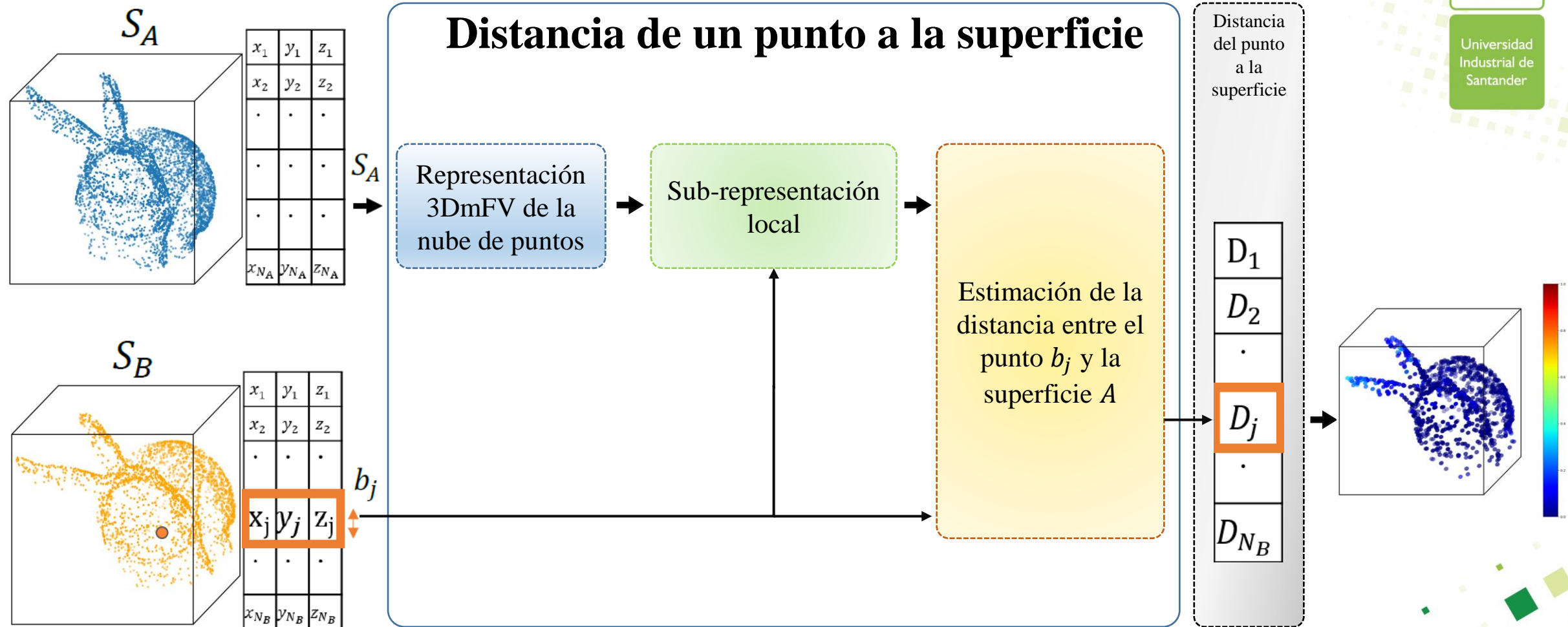
Universidad
Industrial de
Santander



Funcionamiento del modelo DPDist

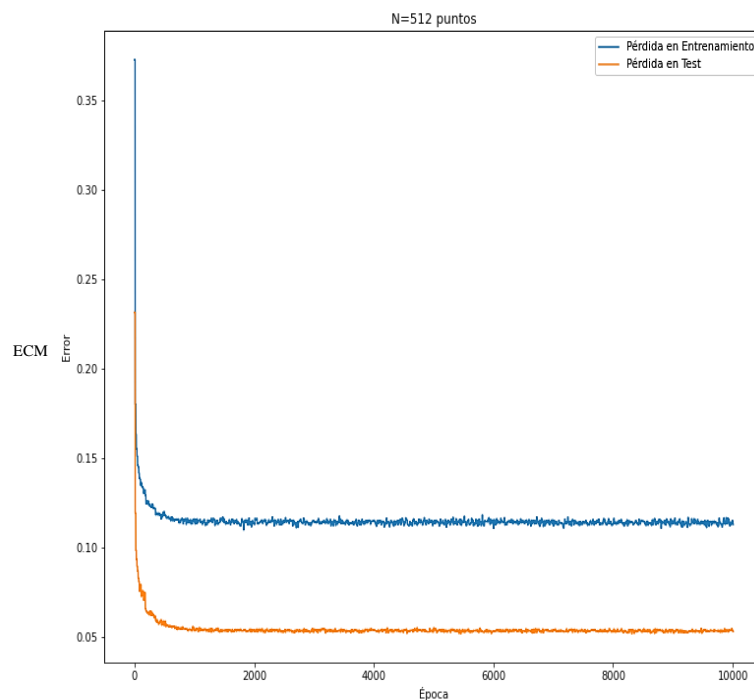


Universidad
Industrial de
Santander

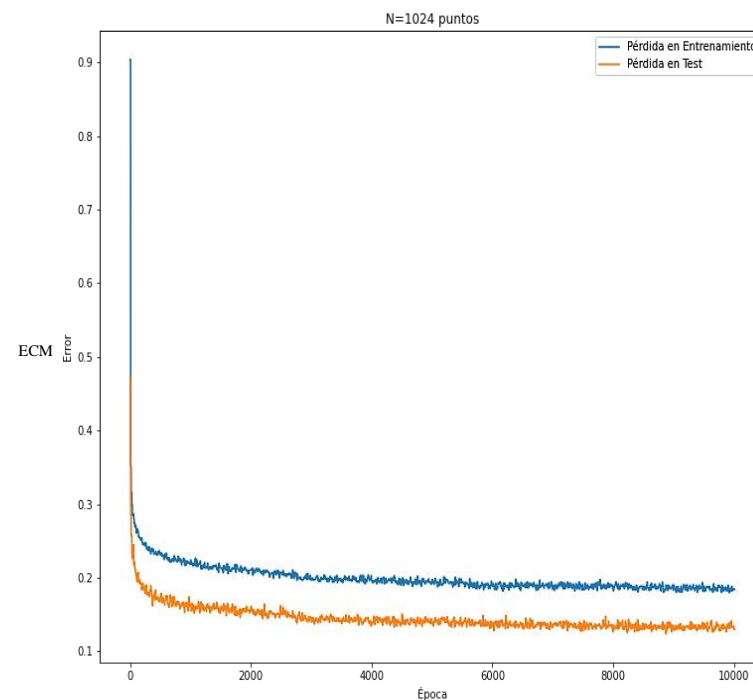




Modelo	Nº de Épocas	Tiempo Total de Entrenamiento	Pérdida/Error Mínimo (entrenamiento)	Pérdida/Error Mínimo (test)
Versión 1	10000	≈ 60 horas, 10 minutos, 47 segundos	≈ 11.317 %	≈ 5.320 %
Versión 2		≈ 21 horas, 8 minutos, 8 segundos	≈ 18.428 %	≈ 12.985 %

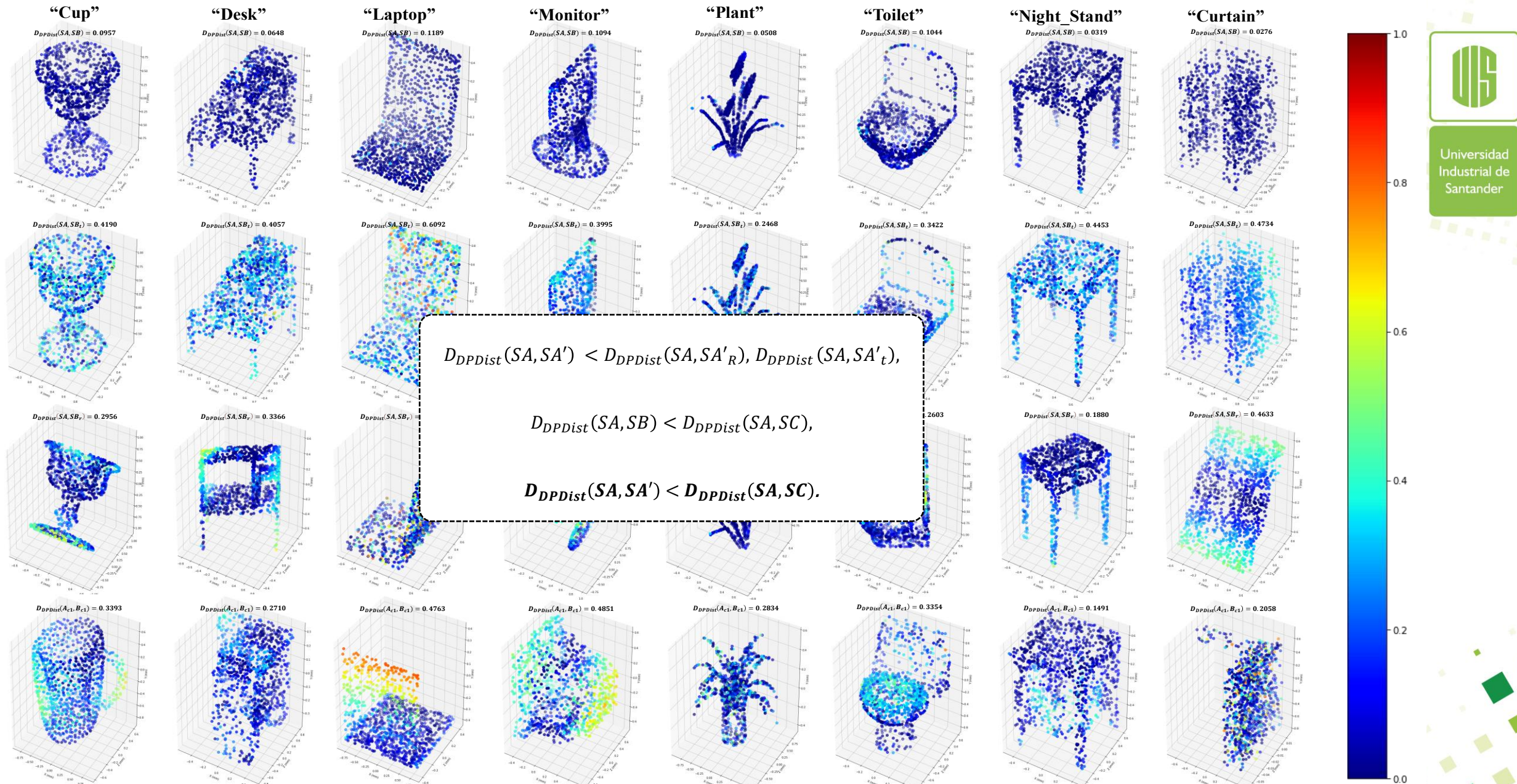


(a)

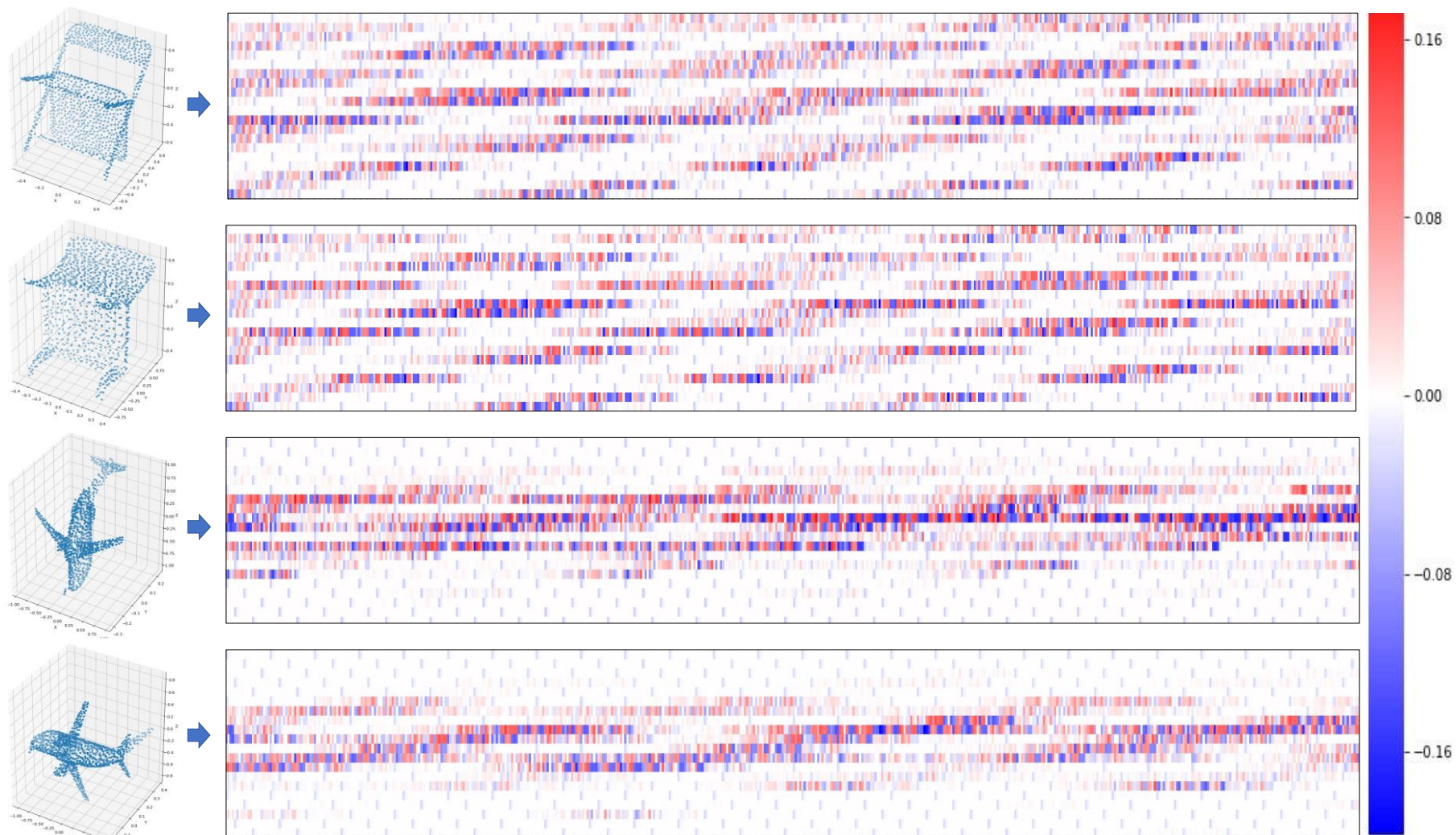


(b)

Resultados del entrenamiento de dos versiones del modelo DPDist: (a) 512 puntos en grupos de 32, (b) 1024 en grupos de 64 puntos.



Resultados de pruebas de comparación de DPDist en distintos escenarios y con diferentes clases de modelos.



Representación 3DmFV para diferentes clases de modelos, usando 8^3 gaussianas.



Algoritmo Propuesto: correspondencia de puntos con gradientes de DPDist

DPDist: Correspondencia de puntos usando retropropagación:

Se propuso usar la métrica de comparación DPDist entre las dos nubes de puntos a registrar para establecer el “matching” o correspondencia de puntos y posteriormente, mediante el uso de la Descomposición en Valores Singulares (en inglés SVD), generar la matriz de rotación implícita entre los dos conjuntos de puntos cuyos puntos son coincidentes.

Correspondencia de puntos con gradientes

DPDist: Retropropagación y gradientes:

Como DDPdist intrínsecamente localiza los puntos cuyo valor de distancia de comparación es mínimo, al realizar la diferenciación automática entre la predicción y los datos de entrada, se obtiene el valor del gradiente que apunta hacia la dirección en la que el DDPdist se minimiza para cada punto.

$$\text{grads} = \text{tf.gradients}(t_{\text{salida}}, t_{\text{entrada}})$$



Proceso de diferenciación automática

1. Se recorre el grafo en forward para guardar los valores de cada variable y las dependencias entre nodos, y obtener la predicción o salida de la red.
2. Se realiza el proceso de backward, empezando por la salida, a partir de la función de costo y calculando las derivadas de cada variable respecto a las demás variables que le anteceden.
3. El proceso termina cuando se alcanzan los nodos que contienen los datos de entrada (las nubes de puntos), obteniendo el vector gradiente para hallar la dirección de las correspondencias de cada punto en la nube.

Estimación de correspondencias

Sean S_A y S_B dos muestras de nubes de puntos a registrar.

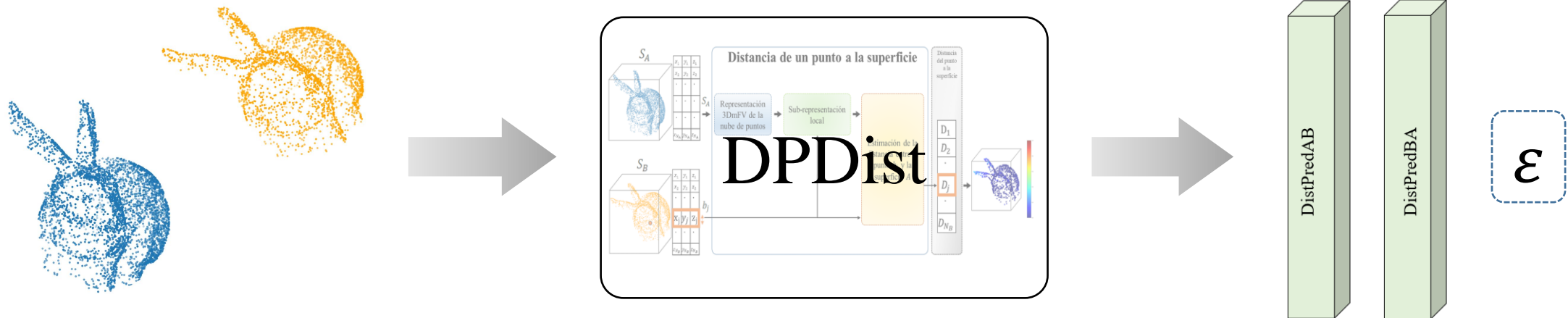


Estimación de correspondencias

Estimación de la distancia de comparación DPDist entre S_A y S_B y el error de predicción.

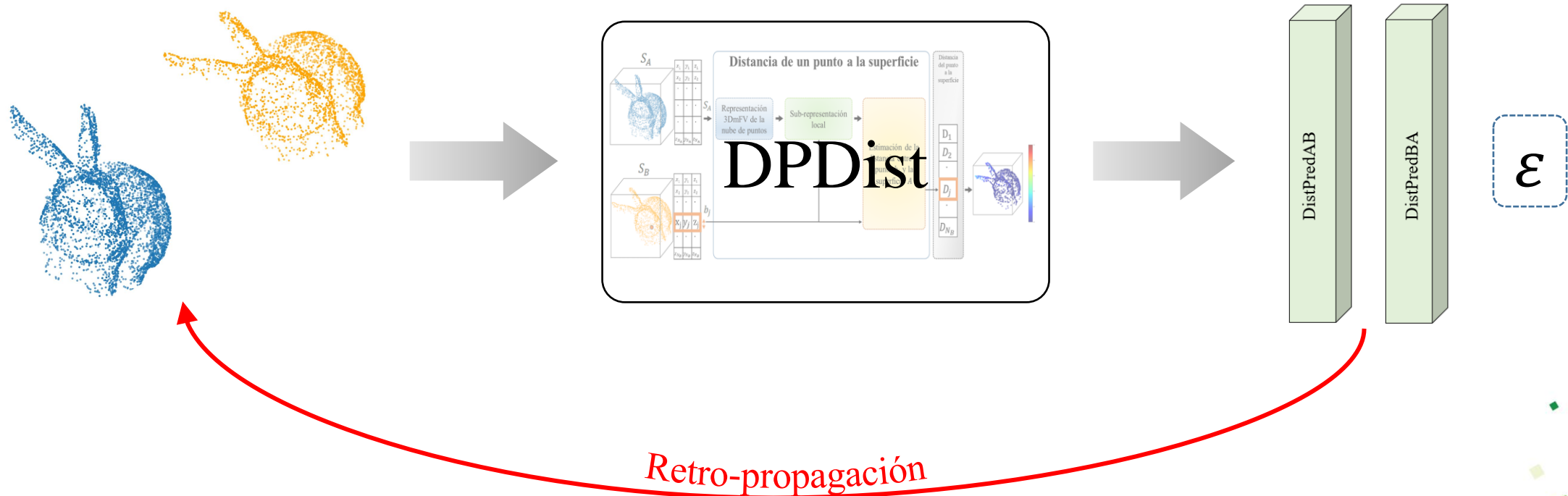


Universidad
Industrial de
Santander



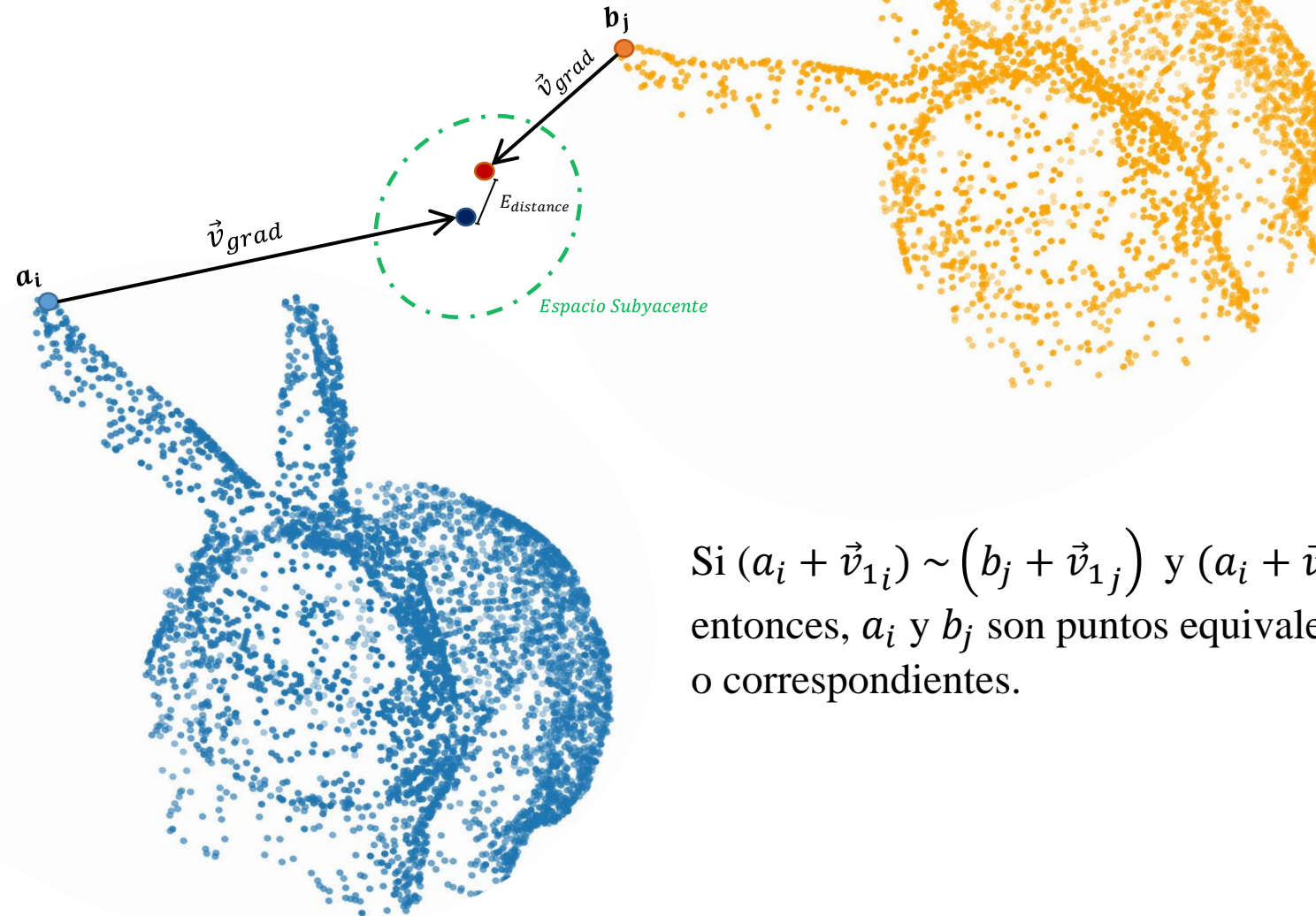
Estimación de correspondencias

Cálculo de los gradientes en cada punto mediante la retro-propagación del DPDist.



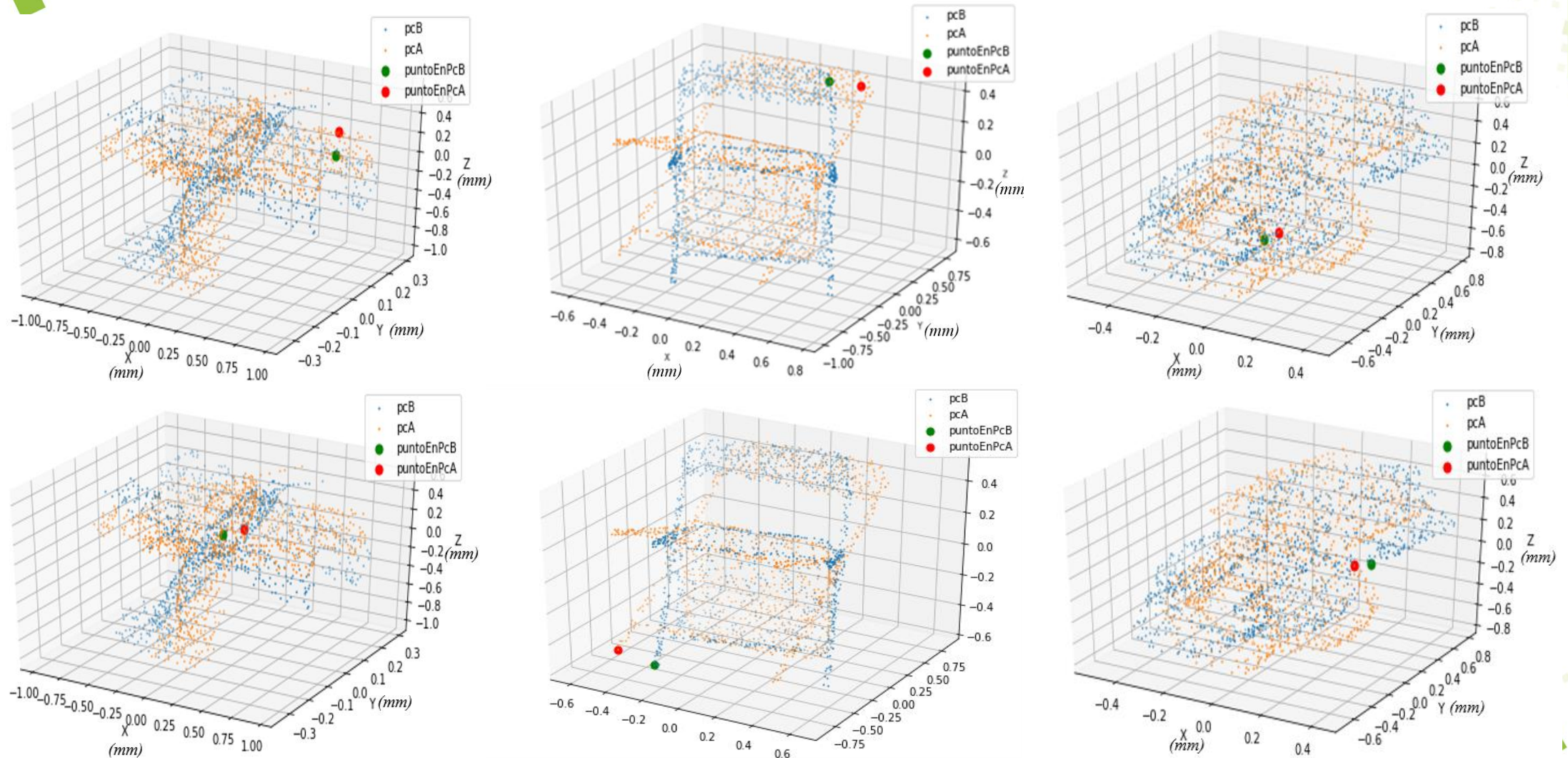
Estimación de correspondencias

$$\vec{v}_{grad} = \begin{cases} \vec{v}_1 \\ \vec{v}_2 \end{cases}$$



Si $(a_i + \vec{v}_{1i}) \sim (b_j + \vec{v}_{1j})$ y $(a_i + \vec{v}_{2i}) \sim (b_j + \vec{v}_{2j})$, entonces, a_i y b_j son puntos equivalentes, coincidentes o correspondientes.

Resultados del emparejamiento (correspondencia) de puntos



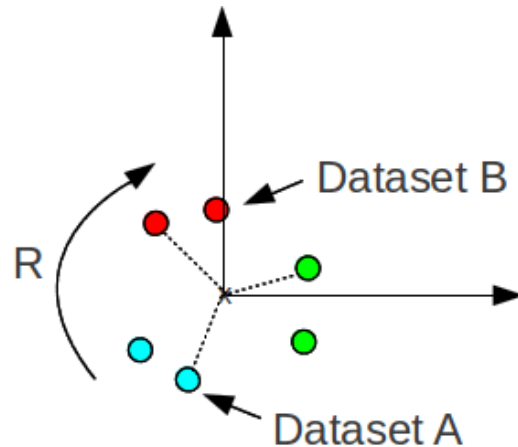


Experimento 1			Experimento 2		Experimento 3	
Iteración	N° de pares de puntos coincidentes	Error de Registro	N° de pares de puntos coincidentes	Error de Registro	N° de pares de puntos coincidentes	Error de Registro
1	26	0.029	29	0.034	20	0.064
2	24	0.033	26	0.037	14	0.066
3	26	0.033	20	0.034	18	0.067
4	28	0.033	30	0.033	13	0.064
5	32	0.032	30	0.032	25	0.067
6	35	0.031	31	0.053	21	0.063
7	31	0.032	31	0.042	23	0.066
8	35	0.031	41	0.043	21	0.065
9	36	0.030	35	0.040	25	0.067
10	35	0.030	33	0.043	22	0.065
Promedio	30.4	0.031	30.6	0.040	20.6	0.065

Tabla de resultados de pruebas de predicción de puntos coincidentes entre nubes de puntos a registrar.

Registro de nubes de puntos usando los puntos correspondientes.

A partir de dos nubes de puntos que comparten puntos correspondientes, es posible usar la descomposición de valores singulares para estimar la matriz de transformación rígida.



$$[U, S, V^T] = \text{SVD}(H),$$

$$H = USV^T,$$

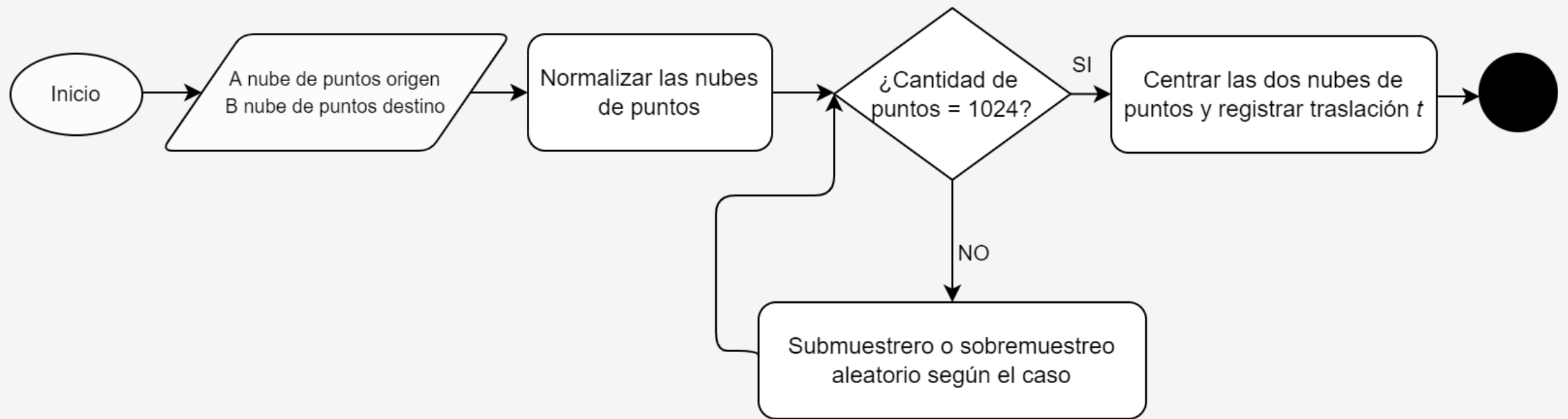
$$R = VU^T.$$

$$t = \text{centroide}_B - R \cdot \text{centroide}_A.$$

$$T = \begin{pmatrix} R_{1,1} & R_{1,2} & R_{1,3} & t_1 \\ R_{2,1} & R_{2,2} & R_{2,3} & t_2 \\ R_{3,1} & R_{3,2} & R_{3,3} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

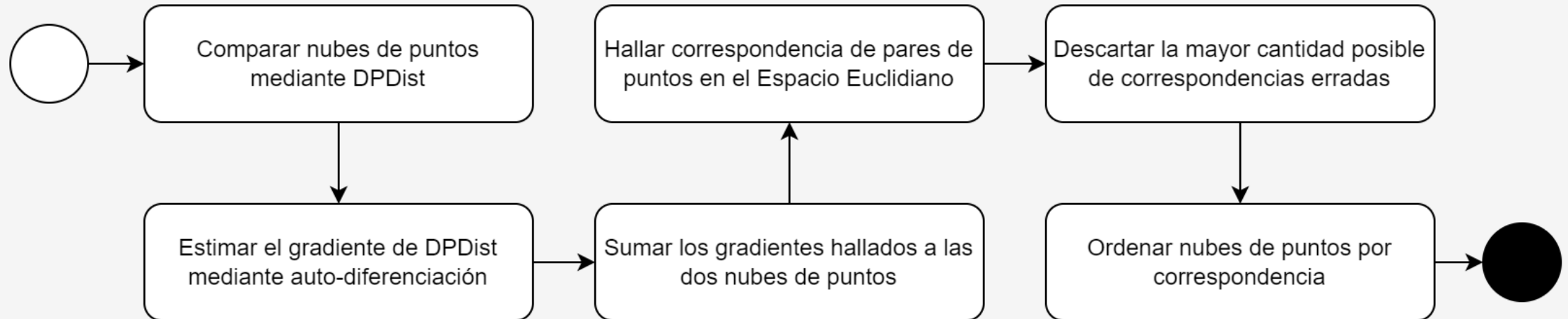
Algoritmo Propuesto

Pretratamiento de datos

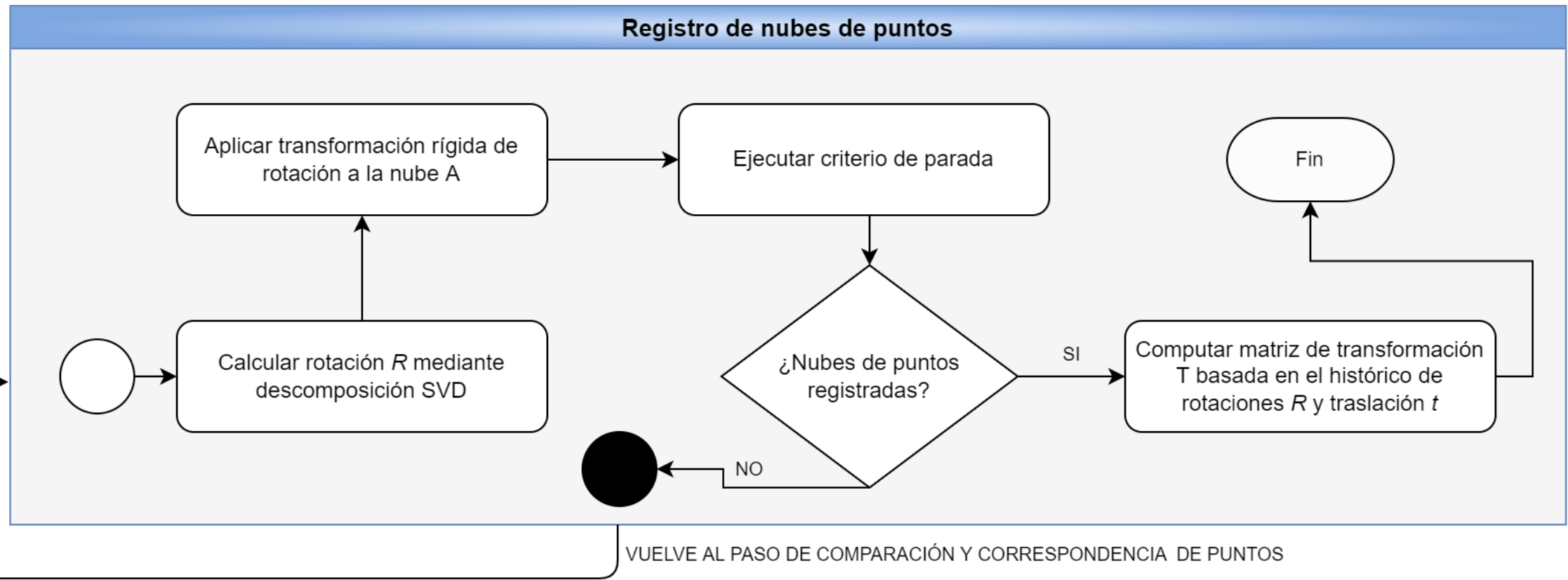


Algoritmo Propuesto

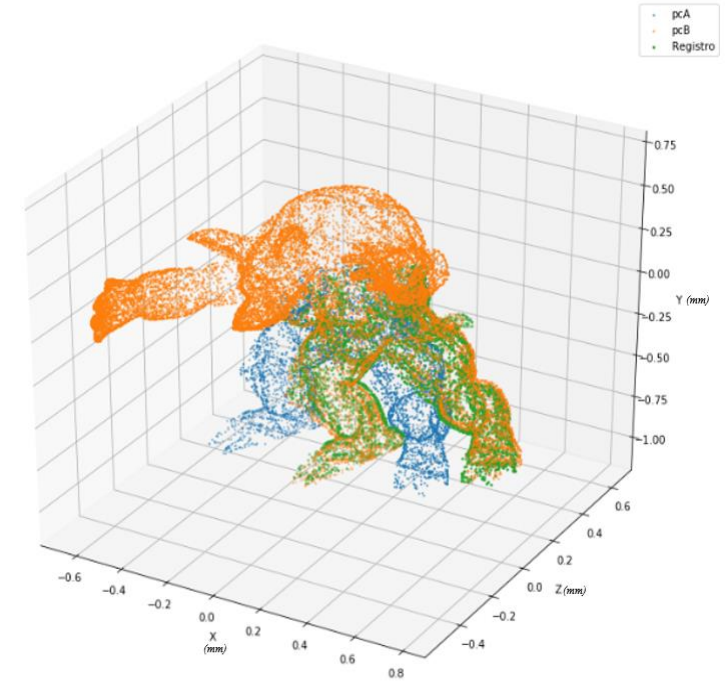
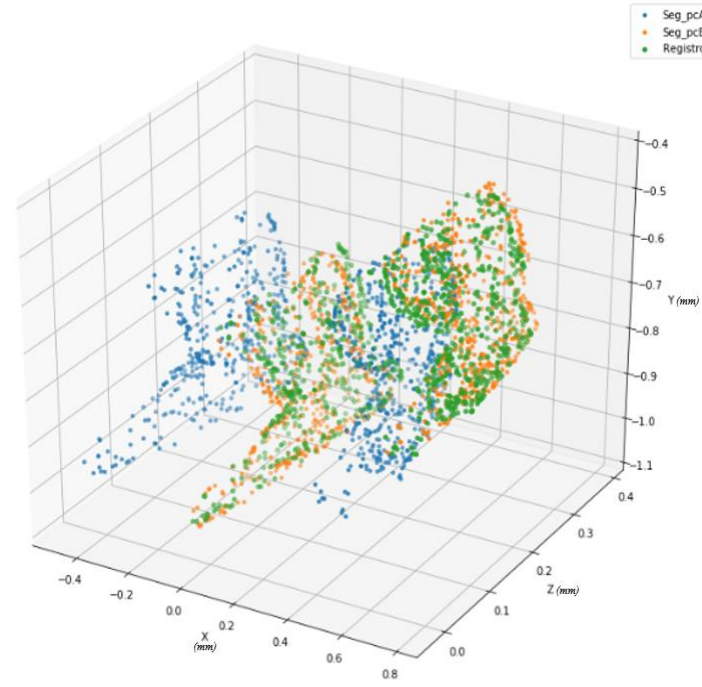
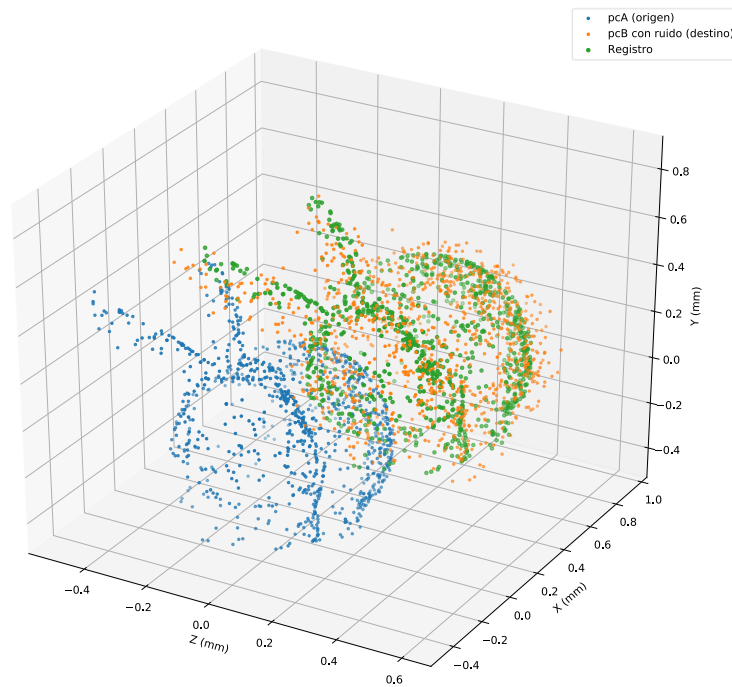
Comparación y correspondencia de puntos



Algoritmo Propuesto



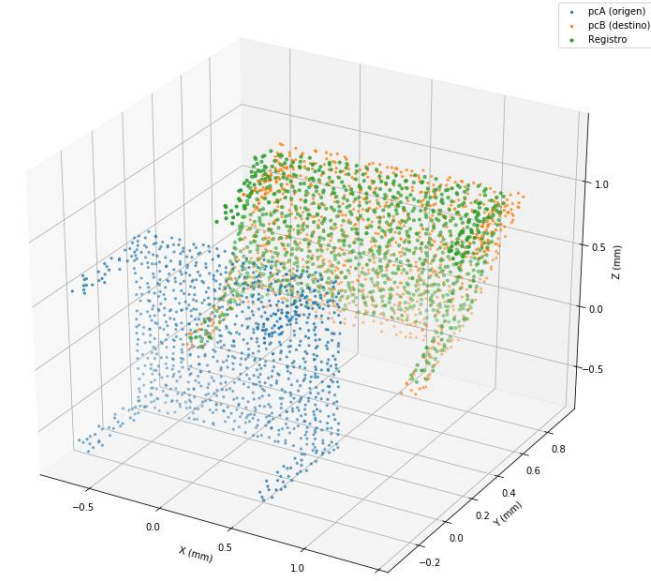
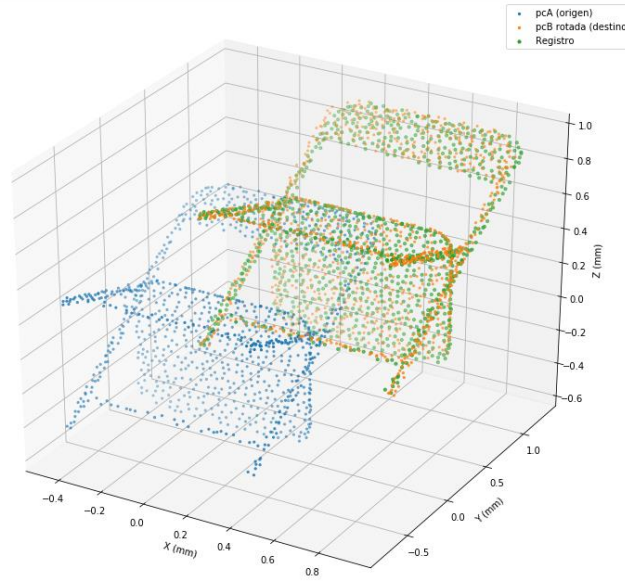
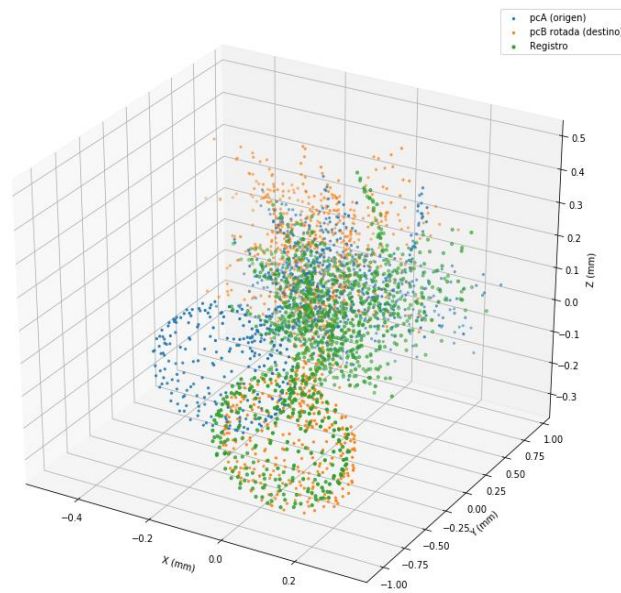
Resultados



Resultados de pruebas de registro de nubes de puntos usando el Algoritmo Propuesto.



Resultados



Resultados de pruebas de registro de nubes de puntos usando el Algoritmo Propuesto.



Resultados

		Experimento 1				Experimento 2				Experimento 3				Experimento 4			
Ruido[u]	iter	E_1 [%]	E_2 [%]	E_3 [%]	E_4 [%]	E_1 [%]	E_2 [%]	E_3 [%]	E_4 [%]	E_1 [%]	E_2 [%]	E_3 [%]	E_4 [%]	E_1 [%]	E_2 [%]	E_3 [%]	E_4 [%]
0.02	5	0.073	0.047	0.277	7.331	0.012	0.012	0.292	8.807	0.025	0.070	0.319	6.931	0.105	0.040	0.320	6.903
	10	0.051	0.001	0.324	12.73	0.103	0.093	0.325	18.19	0.162	0.018	0.320	15.28	0.075	0.062	0.343	13.70
	15	0.016	0.044	0.323	11.54	0.081	0.025	0.383	10.53	0.130	0.170	0.376	16.08	0.110	0.033	0.409	12.40
0.03	5	0.070	0.174	0.281	6.564	0.228	0.177	0.297	9.261	0.244	0.213	0.315	7.276	0.286	0.195	0.323	6.871
	10	0.213	0.085	0.329	12.16	0.233	0.176	0.353	15.58	0.146	0.415	0.539	14.01	0.251	0.066	0.342	12.15
	15	0.050	0.166	0.470	11.43	0.198	0.109	0.284	15.67	0.190	0.187	0.374	14.95	0.174	0.009	0.404	13.91
0.04	5	0.524	0.443	0.278	5.778	0.371	0.382	0.296	8.400	0.303	0.517	0.320	6.788	0.330	0.363	0.325	6.774
	10	0.255	0.387	0.320	12.01	0.417	0.245	0.353	15.91	0.286	0.374	0.351	13.25	0.462	0.596	0.352	10.89
	15	0.362	0.438	0.363	11.26	0.271	0.318	0.378	17.44	0.330	0.382	0.380	10.46	0.383	0.529	0.387	14.09
0.05	5	0.682	0.554	0.288	5.204	0.670	0.542	0.309	6.781	0.498	0.560	0.317	7.199	0.543	0.708	0.323	6.400
	10	0.352	0.569	0.326	18.72	0.490	0.700	0.357	13.75	0.435	0.454	0.329	14.94	0.796	0.589	0.356	11.19
	15	0.660	0.567	0.541	11.56	0.255	0.464	0.372	15.82	0.403	0.333	0.396	14.18	0.609	0.475	0.416	10.63
Promedio		0.275	0.289	0.343	10.52	0.277	0.270	0.333	13.01	0.262	0.307	0.360	11.77	0.343	0.305	0.358	10.49

E_1 : Error relativo del algoritmo propuesto.
 E_2 : Error relativo de ICP.
 E_3 : Error relativo de RANSAC.
 E_4 : Error relativo de CPD.

Tabla de resultados de pruebas de registro de nubes de puntos en comparación con otros algoritmos.



Resultados

		Experimento 1				Experimento 2				Experimento 3				Experimento 4			
Ruido[u]	iter	T_1 [s]	T_2 [s]	T_3 [s]	T_4 [s]	T_1 [s]	T_2 [s]	T_3 [s]	T_4 [s]	T_1 [s]	T_2 [s]	T_3 [s]	T_4 [s]	T_1 [s]	T_2 [s]	T_3 [s]	T_4 [s]
0.02	5	32.63	0.229	0.029	55.24	30.28	0.129	0.001	58.13	32.24	0.134	0.002	58.03	30.78	0.123	0.001	57.92
	10	63.68	0.224	0.002	52.16	60.78	0.219	0.004	51.74	64.95	0.208	0.005	51.70	57.57	0.224	0.007	52.26
	15	99.98	0.371	0.018	59.83	96.89	0.294	0.020	52.88	116.2	0.289	0.014	61.05	95.88	0.323	0.005	60.28
0.03	5	35.21	0.138	0.001	59.26	34.23	0.128	0.001	58.16	36.98	0.122	0.001	57.82	34.67	0.117	0.004	57.78
	10	70.56	0.217	0.010	51.80	71.53	0.217	0.005	51.81	74.86	0.205	0.002	52.05	74.85	0.212	0.023	52.30
	15	107.1	0.277	0.010	59.91	98.77	0.319	0.040	60.81	106.1	0.298	0.015	60.66	110.3	0.273	0.032	60.45
0.04	5	34.48	0.149	0.004	58.82	34.45	0.130	0.001	57.01	37.63	0.131	0.001	57.65	37.46	0.125	0.003	37.75
	10	77.56	0.264	0.002	51.76	71.79	0.218	0.006	51.69	75.81	0.247	0.015	52.77	79.36	0.257	0.016	52.50
	15	122.1	0.330	0.013	59.51	118.0	0.303	0.031	60.34	114.7	0.321	0.008	54.27	129.8	0.354	0.007	59.64
0.05	5	38.75	0.163	0.001	59.60	38.17	0.122	0.015	58.45	39.84	0.130	0.001	57.59	40.79	0.125	0.002	38.02
	10	84.39	0.203	0.004	58.07	75.76	0.240	0.021	51.53	82.48	0.257	0.005	52.39	84.65	0.252	0.004	52.46
	15	128.9	0.295	0.015	59.84	121.1	0.318	0.012	68.13	133.6	0.291	0.011	60.26	135.1	0.379	0.023	59.97
Promedio		78.43	0.238	0.009	57.150	70.99	0.220	0.013	56.72	76.30	0.219	0.007	56.35	75.95	0.230	0.011	53.44

T_1 : Tiempo de ejecución del algoritmo propuesto.

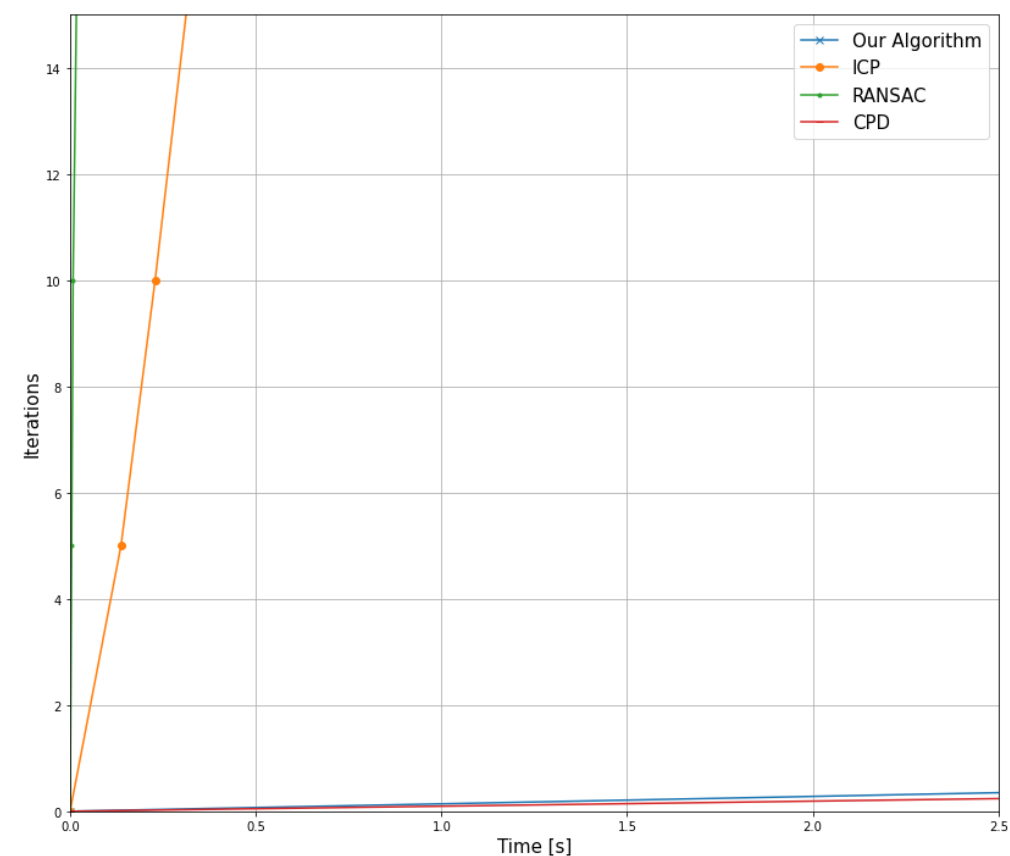
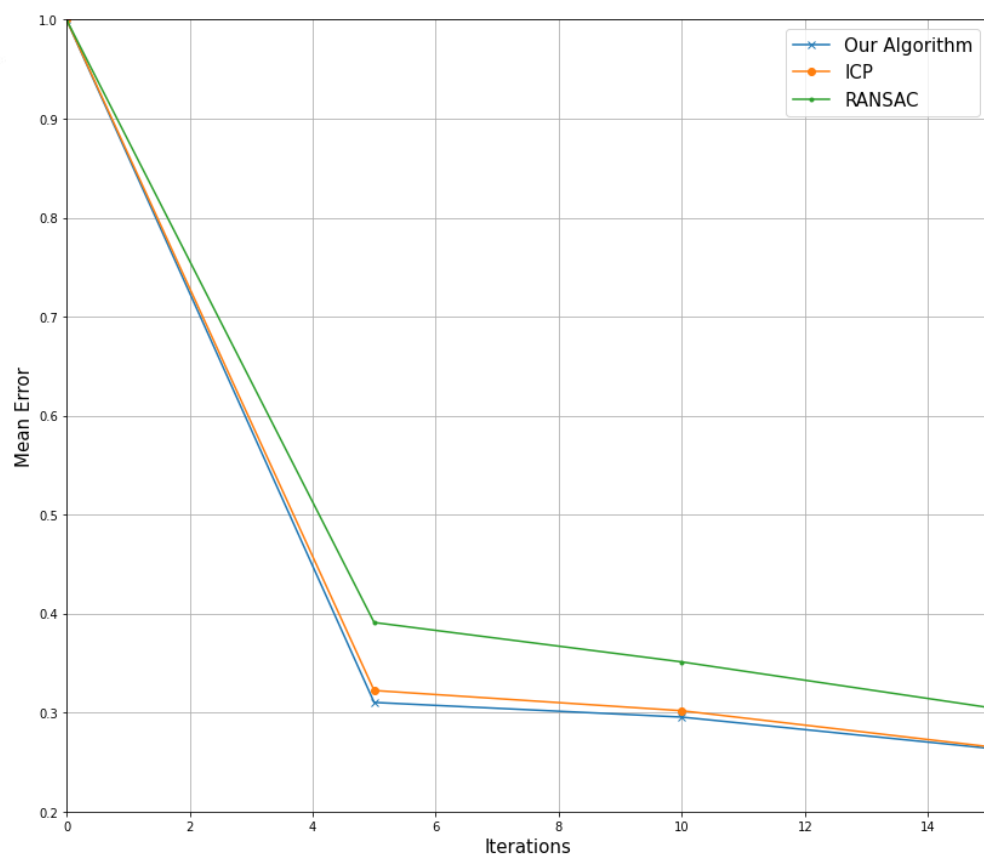
T_2 : Tiempo de ejecución de ICP.

T_3 : Tiempo de ejecución de RANSAC.

T_4 : Tiempo de ejecución de CPD.

Tabla de resultados de tiempos de ejecución para el registro de nubes de puntos en comparación con otros algoritmos.

Resultados



Graficas de comparación de exactitud en el registro y tiempos de ejecución en comparación con otros algoritmos.

Conclusiones

El algoritmo propuesto no necesita una estimación inicial de correspondencia de puntos ni de la intervención humana para hacerlo, sino que automáticamente estima los pares de puntos coincidentes en las dos nubes de puntos a registrar.

El algoritmo propuesto sacrifica tiempo de ejecución, ya que en base a los resultados es muy lento en comparación con otros algoritmos como ICP, RANSAC o CPD. Los largos tiempos obtenidos en el algoritmo propuesto se deben al costo computacional para el cálculo o estimación de los gradientes mediante la diferenciación automática en cada iteración del algoritmo.

Conclusiones

A nivel de software, el algoritmo propuesto cumple con los requerimientos establecidos pues permite un óptimo registro de nubes de puntos tridimensionales (cuantificado por el error relativo de resultados teóricos y experimentales) mediante el uso de técnicas de Aprendizaje Profundo. Además, satisface medidas de exactitud a partir del error relativo respecto a registros hechos manualmente y es capaz de competir con otros algoritmos de registro vigentes en el estado del arte.

Trabajo Futuro / Recomendaciones

Mejorar la estimación de correspondencia o coincidencia de puntos es clave para mejorar el rendimiento del algoritmo propuesto, esto se podría lograr mediante:

- Un entrenamiento más robusto del DPDist.
- La modificación del modelo DPDist para que además de generar una representación por localidades, no pierda la generalidad de la nube de puntos.

Trabajo Futuro / Recomendaciones

Posibilidad de usar una arquitectura de Red Neuronal Recurrente para tratar los datos secuenciales y poder predecir los ángulos de rotación implícitos entre dos nubes de puntos comparadas mediante DPDist.

Uso del método propuesto para el hallazgo de correspondencias de puntos mediante el gradiente del modelo DPDist, en combinación con algoritmos de registro como RANSAC, CPD y otros.

Eventos Académicos

El presente trabajo de investigación fue presentado como ponencia mediante póster en el evento internacional *Latin America Optics and Photonics Conference**, organizado por la Sociedad de Óptica Estadounidense (OSA) y llevado a cabo en la ciudad de Recife, Pernambuco, Brasil, del día 07 al 11 de agosto del año en curso.





GitHub del proyecto



Universidad
Industrial de
Santander

#LaUISqueQueremos

iGracias!