



DeepSDF

LERMUSIAUX Nicolas, OURKIA Abdelhakim, PERRIN Killian, SARHANE Abdelmouhaimen

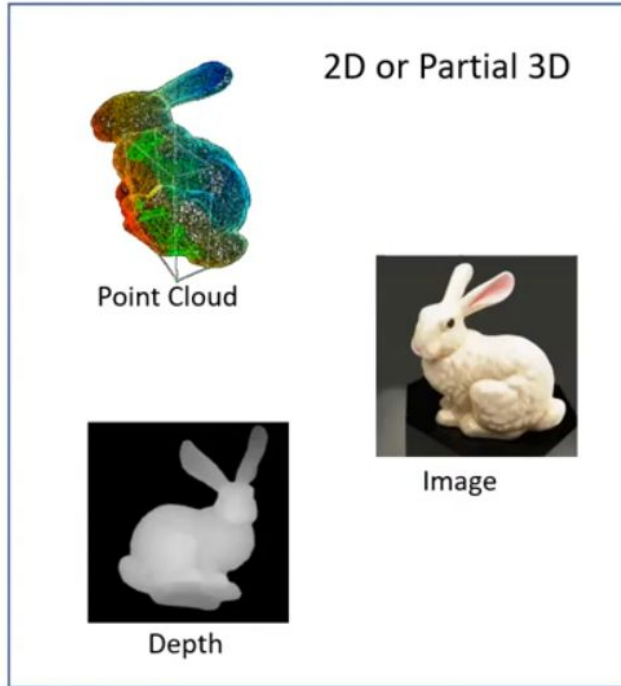
Sommaire



- Motivations et Intérêt du papier
- Modèle
- Résultats et Metrics employés

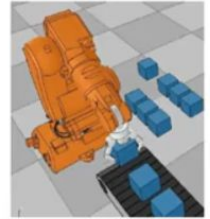
Motivations

- Reconstruction 3D
- Domaines d'application : robotique, simulation
- Objectif : trouver une représentation 3D compact et expressive



Reconstruction

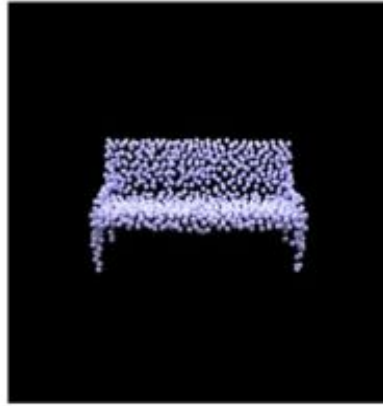
Graphics



Motivations



Voxel (Choy et al. 2016)



Point Cloud (Fan et al. 2017)



Mesh (Groueix et al. 2017)

Parmi les types de représentation 3D existantes : voxels, maillage, nuage de points

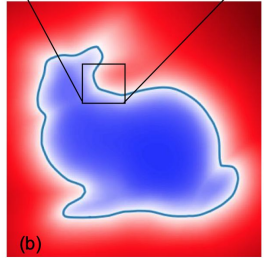
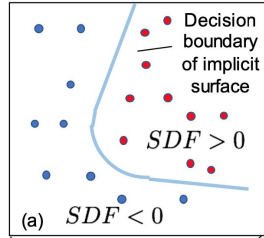
-> **nuage de points** : on connaît les coordonnées d'un ensemble de point mais pas la manière dont ils sont reliés

-> pas d'information sur la topologie de l'objet !

-> **voxels** : coûteux en termes de calcul et de mémoire.

-> **maillages** : limités par une topologie fixée et tendent à masquer les artefacts.

Intérêt du papier




(c)



- Régression directe de la SDF
- La surface est représentée implicitement par le niveau zéro de la fonction.

$$f_{\theta}(x, y, z) \approx \text{SDF}(x, y, z)$$

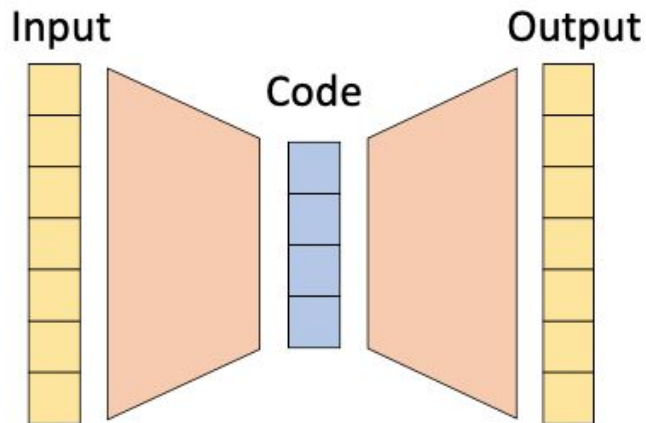
Intérêt du papier



Method	Type	Discretization	Complex topologies	Closed surfaces	Surface normals	Model size (GB) (s)	Inf. time (s)	Eval. tasks
3D-EPN [16]	Voxel SDF	32^3 voxels	✓	✓	✓	0.42	-	C
OGN [52]	Octree	256^3 voxels	✓	✓		0.54	0.32	K
AtlasNet-Sphere [22]	Parametric mesh	1 patch		✓		0.015	0.01	K, U
AtlasNet-25 [22]	Parametric mesh	25 patches	✓			0.172	0.32	K, U
DeepSDF (ours)	Continuous SDF	none	✓	✓	✓	0.0074	9.72	K, U, C

Table 1: Overview of the benchmarked methods. AtlasNet-Sphere can only describe topological-spheres, voxel/octree occupancy methods (i.e. OGN) only provide 8 directions for normals, and AtlasNet does not provide oriented normals. Our tasks evaluated are: (K) representing known shapes, (U) representing unknown shapes, and (C) shape completion.

Auto-encoder pour estimer la SDF

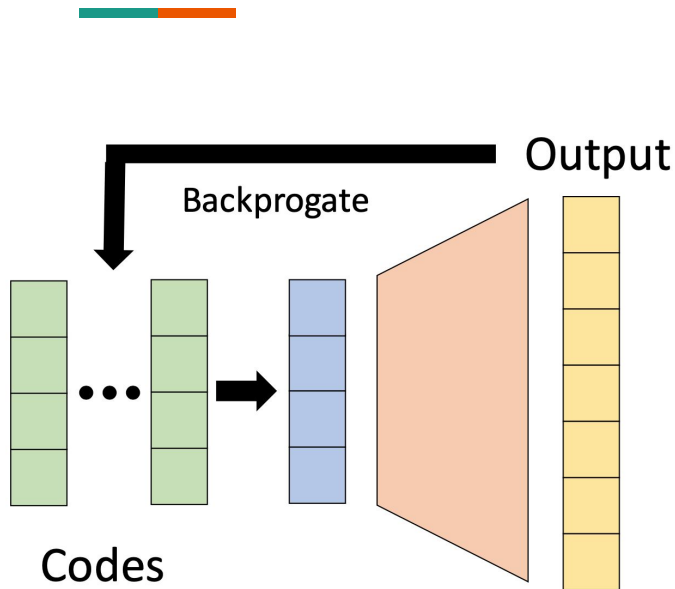


- Objectifs : représenter une grande variété de formes, découvrir leurs propriétés communes et les intégrer dans un espace latent de faible dimension
- On apprend un espace latent optimal qui capture les caractéristiques essentielles des données.

Problème : L'encodeur peut être une perte de ressources calculs pendant l'entraînement.

Les auteurs ont choisi de supprimer l'encodeur

Auto-décodeur pour estimer la SDF



Entraînement : optimise simultanément :

- les vecteurs latents associé à chaque objet 3D
- les poids du décodeur par rétropropagation.

Inférence : un vecteur latent optimal est recherché afin de correspondre à la nouvelle observation avec des paramètres de décodeur fixes.

Adaptée à des problématiques comme la réduction de bruit ou la reconstruction de mesures manquantes

Quelle forme ont les entrées ?

- Partant d'une représentation 3D, on échantillonne un ensemble de point 3D dont on calcule la SDF

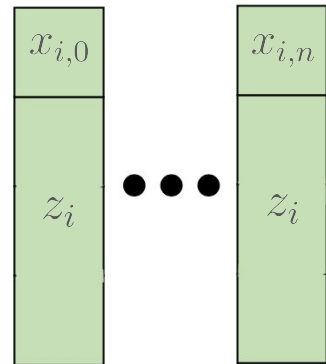


$$X := \{(\mathbf{x}, s) : SDF(\mathbf{x}) = s\}.$$

- Notre ensemble de données d'entraînement est donc un ensemble de forme :

$$X_i = \{(\mathbf{x}_j, s_j) : s_j = SDF^i(\mathbf{x}_j)\}$$

- Formation du vecteur d'entrée : $(X_i, z_i)_{i \in \llbracket 1; N \rrbracket}$



Choix de la fonction coût

- Chacun des vecteurs latents en entrée est fourni au decoder

- 
- Estimation de la valeur par la SDF

$$\tilde{s}_j = f_{\theta}(\mathbf{z}_i, \mathbf{x}_j)$$

- Expression de la vraisemblance de la SDF :

$$p_{\theta}(\mathbf{s}_j | \mathbf{z}_i; \mathbf{x}_j) = \exp(-\mathcal{L}(f_{\theta}(\mathbf{z}_i, \mathbf{x}_j), s_j))$$

- Le posterior qui représente la probabilité du code de forme \mathbf{z}_i donné les échantillons SDF X_i , peut être exprimé simplement comme suit :

$$p_{\theta}(\mathbf{z}_i | X_i) = p(\mathbf{z}_i) \prod_{(\mathbf{x}_j, \mathbf{s}_j) \in X_i} p_{\theta}(\mathbf{s}_j | \mathbf{z}_i; \mathbf{x}_j)$$

- Avec $p(\mathbf{z}_i)$ la distribution a priori supposé être une Gaussienne centrée multivariée de covariance $\sigma^2 I$.

Choix de la fonction coût



$$\mathcal{L}(f_{\theta}(\mathbf{x}), s) = | \text{clamp}(f_{\theta}(\mathbf{x}), \delta) - \text{clamp}(s, \delta) |$$

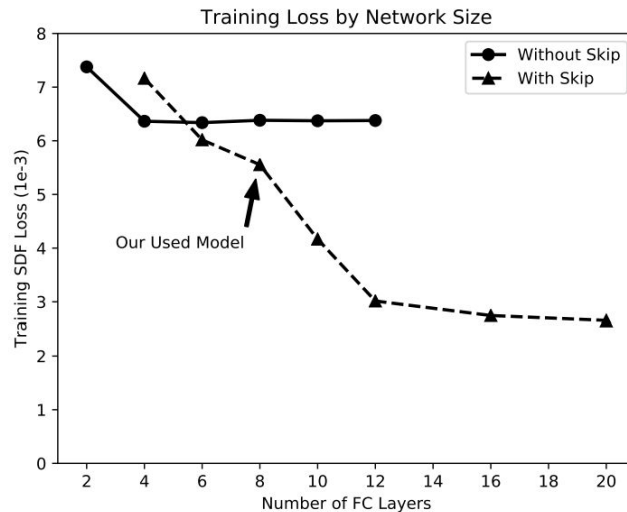
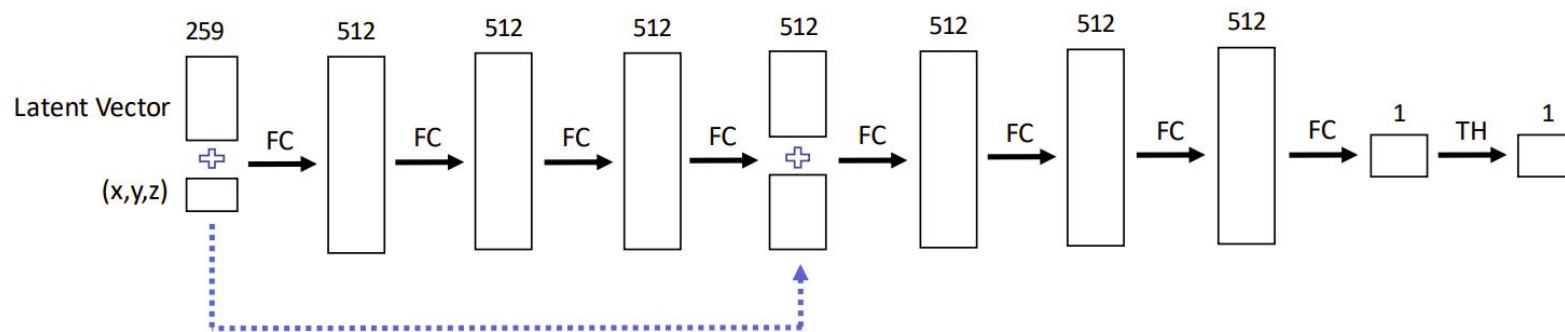
$$\text{clamp}(x, \delta) := \min(\delta, \max(-\delta, x))$$

L'introduction de la fonction **clamp** permet de contrôler, à l'aide du paramètre δ , la distance par rapport à la surface à laquelle on souhaite conserver une métrique SDF précise. En ajustant δ , il est possible de réguler la précision de la reconstruction.

Finalement, on cherche les paramètres θ et $\{\mathbf{z}_i\}_{i=1}^N$ vérifiant :

$$\arg \min_{\theta, \{\mathbf{z}_i\}_{i=1}^N} \sum_{i=1}^N \left(\sum_{j=1}^K \mathcal{L}(f_{\theta}(\mathbf{z}_i, \mathbf{x}_j), s_j) + \frac{1}{\sigma^2} \|\mathbf{z}_i\|_2^2 \right)$$

Architecture utilisée pour DeepSDF




Résultats



Méthode d'évaluation:

- Conversion de DeepSDF en maillage avec Marching Cubes en résolution 512^3
- AtlasNet a ses sommets pondéré par la surface des triangles du maillage
- 3D-EPN a son iso-surface extraite à l'aide de MATLAB


Résultats



Mesh comp., mean	chair	plane	table	lamp	sofa
AtlasNet-Sph.	0.668	0.862	0.755	0.281	0.641
AtlasNet-25	0.723	0.887	0.785	0.528	0.681
DeepSDF	0.947	0.943	0.959	0.877	0.931
Mesh comp., median					
AtlasNet-Sph.	0.686	0.930	0.795	0.257	0.666
AtlasNet-25	0.736	0.944	0.825	0.533	0.702
DeepSDF	0.970	0.970	0.982	0.930	0.941
Cosine sim., mean					
AtlasNet-Sph.	0.790	0.840	0.826	0.719	0.847
AtlasNet-25	0.797	0.858	0.835	0.725	0.826
DeepSDF	0.896	0.907	0.916	0.862	0.917

Pour différentes métriques, l'algorithme **DeepSDF** tend à avoir de bien meilleur résultats que d'autres algorithmes de références

Résultats



Method \ metric	CD, mean	CD, median	EMD, mean	EMD, median
OGN	0.167	0.127	0.043	0.042
AtlasNet-Sph.	0.210	0.185	0.046	0.045
AtlasNet-25	0.157	0.140	0.060	0.060
DeepSDF	0.084	0.058	0.043	0.042

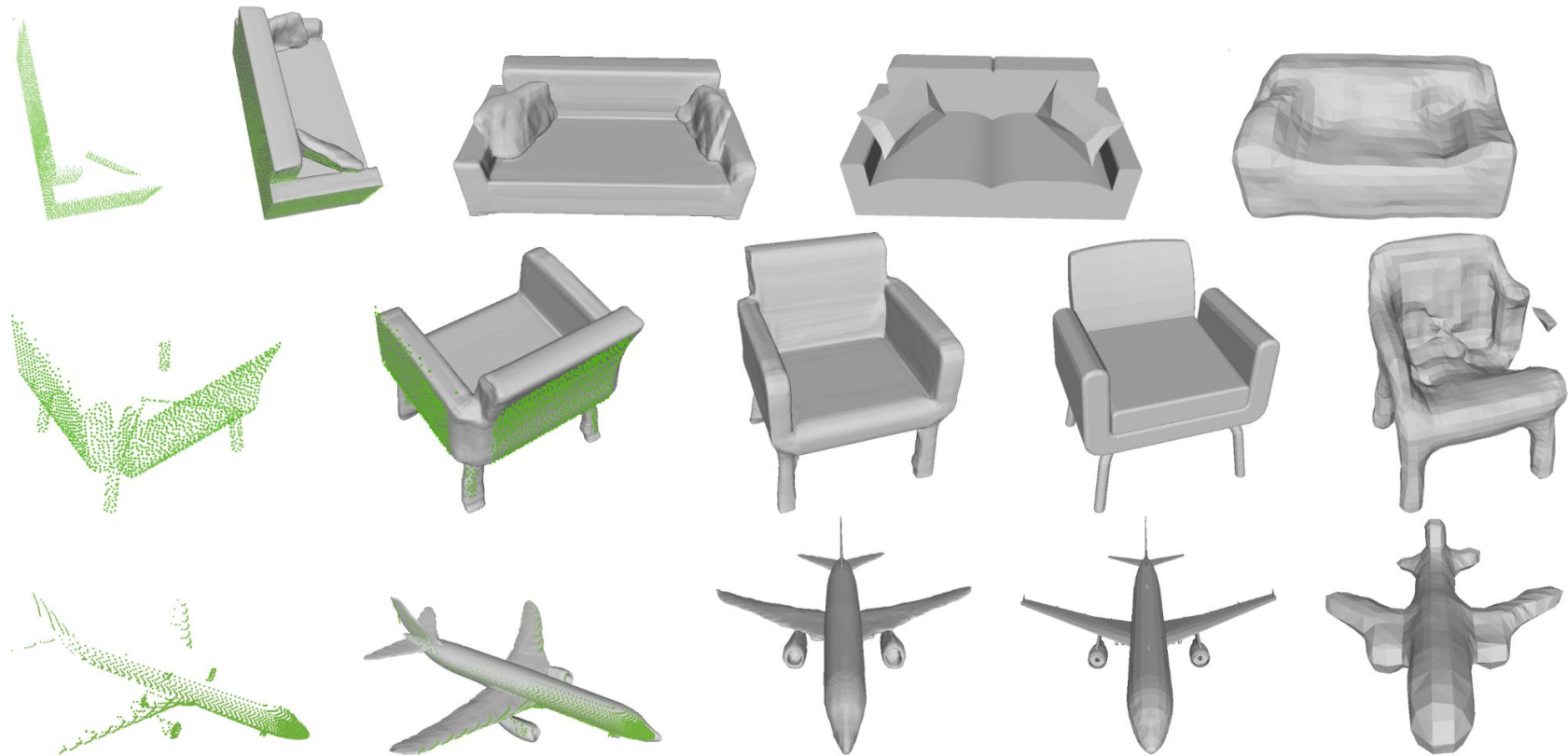
- Distance de Chamfer (CD)

$$d_{CD}(S_1, S_2) = \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2^2$$

- Earth Mover's Distance (EMD)

$$d_{EMD}(S_1, S_2) = \min_{\phi: S_1 \rightarrow S_2} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

Résultats



(a) Input Depth

(b) Completion (ours)

(c) Second View (ours)

(d) Ground truth

(e) 3D-EPN

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation



Learned Chair Shape Space




Learned Car Shape Space

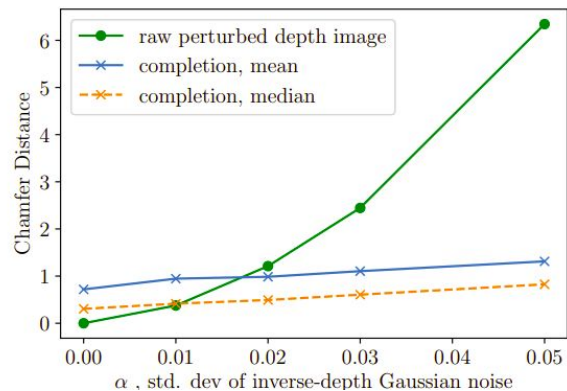
Example Linear Interpolation through DeepSDF latent shape space trained on 4116 chairs and 662 car models



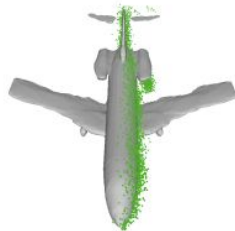
Merci de votre attention

Suppléments

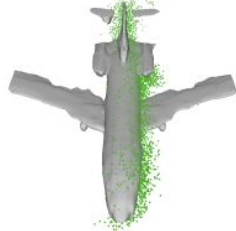

$$D_{\text{noise}} = \frac{1}{(1/D) + \mathcal{N}(0, \alpha^2)}$$



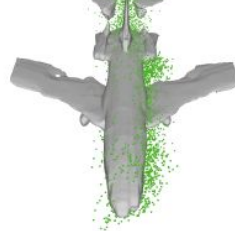
(a) No noise



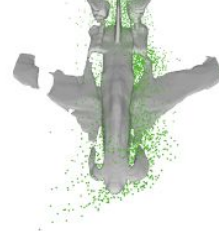
(b) $\alpha = 0.01$



(c) $\alpha = 0.02$



(d) $\alpha = 0.03$



(e) $\alpha = 0.05$