

# KPConv : Flexible and Deformable Convolution for Point Clouds

Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud,  
Beatriz Marcotegui, François Goulette, Leonidas J. Guibas

# Contexte

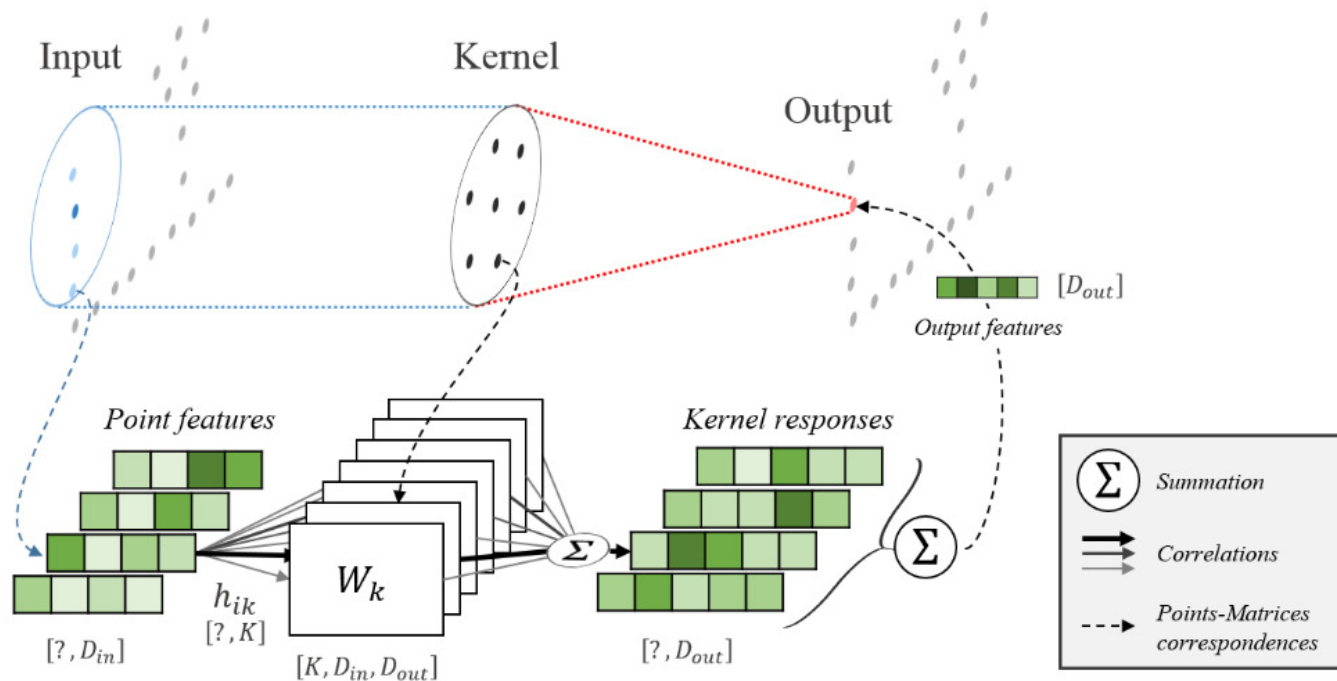
- Article publié en 2019 (CVPR)
- Nuages de points et ML:
  - Utilisation de CNN sur nuages structurés (mesh, voxel, etc.)
  - Approche basé cloud : PointNet (Qi et al., 2017)

=> Manque de flexibilité, réussir à capturer une forme géométrique locale

# Problème posé

- Utiliser des nuages de points
- Répondre aux contraintes:
  - Invariance aux permutations
  - Robuste à certaines transformations géométriques
  - Ensemble non structuré / densité variable

# Noyau de convolution par point



- Voisinage sphérique autour d'un point
- Définition des "cases" de convolution par des points
  - Position des points de conv par système optimal  
=> couverture de la sphère de voisinage
  - point du noyau = Poids + Pondération densité local

$$g(y_i) = \sum_{k < K} h(y_i, \tilde{x}_k) W_k$$

$y_i$  = point du voisinage centré

$g(y_i)$  = kernel fonction pour chaque  $y_i$

$\tilde{x}_k$  = point du kernel

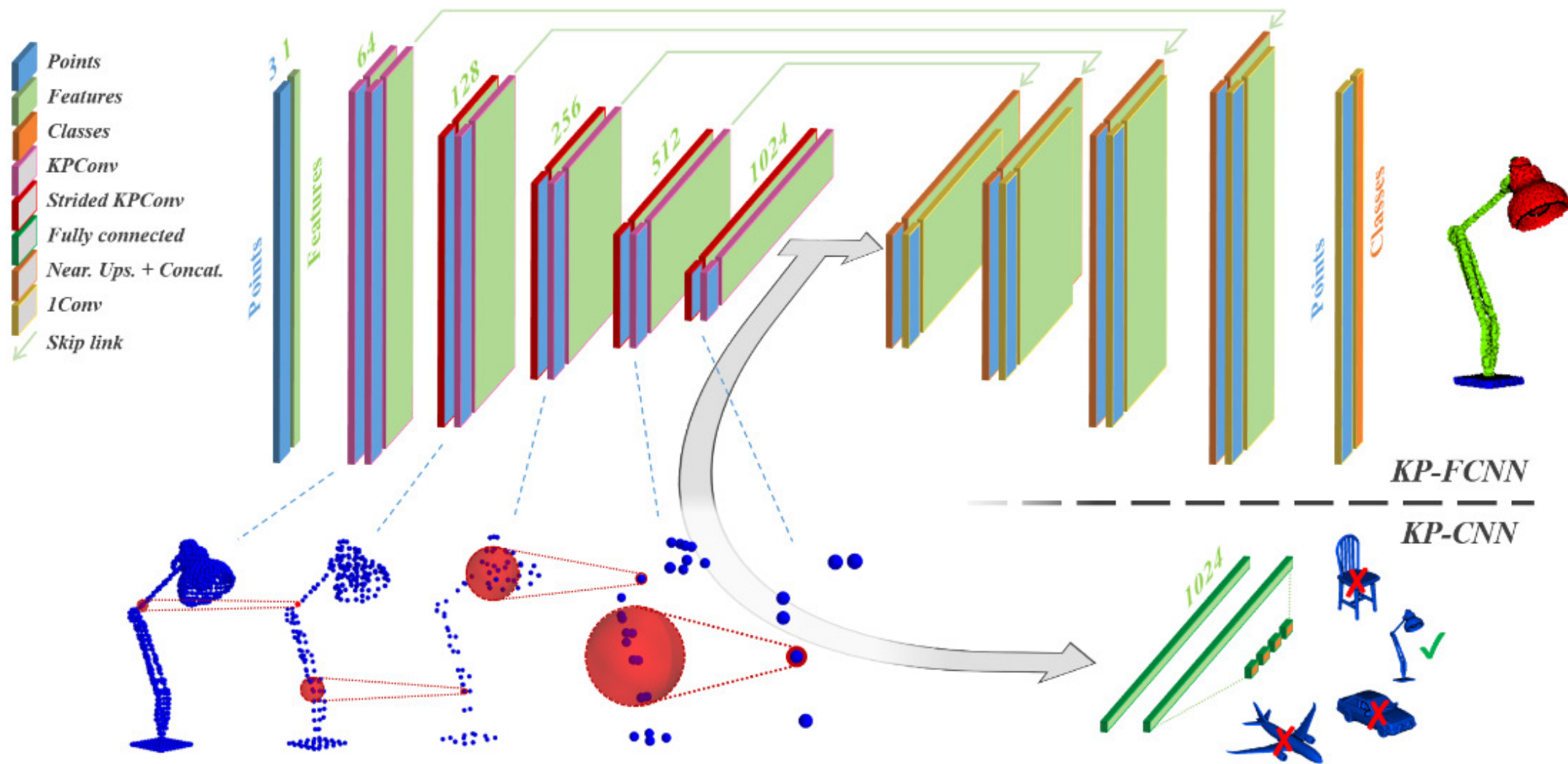
$W_k$  = poids associé au point

$h(y_i, \tilde{x}_k)$  = corrélation entre point  $y_i$  et le point du kernel  $\tilde{x}_k$

$$h(y_i, \tilde{x}_k) = \max \left( 0, 1 - \frac{\|y_i - \tilde{x}_k\|}{\sigma} \right)$$

$\sigma$  = densité du nuage

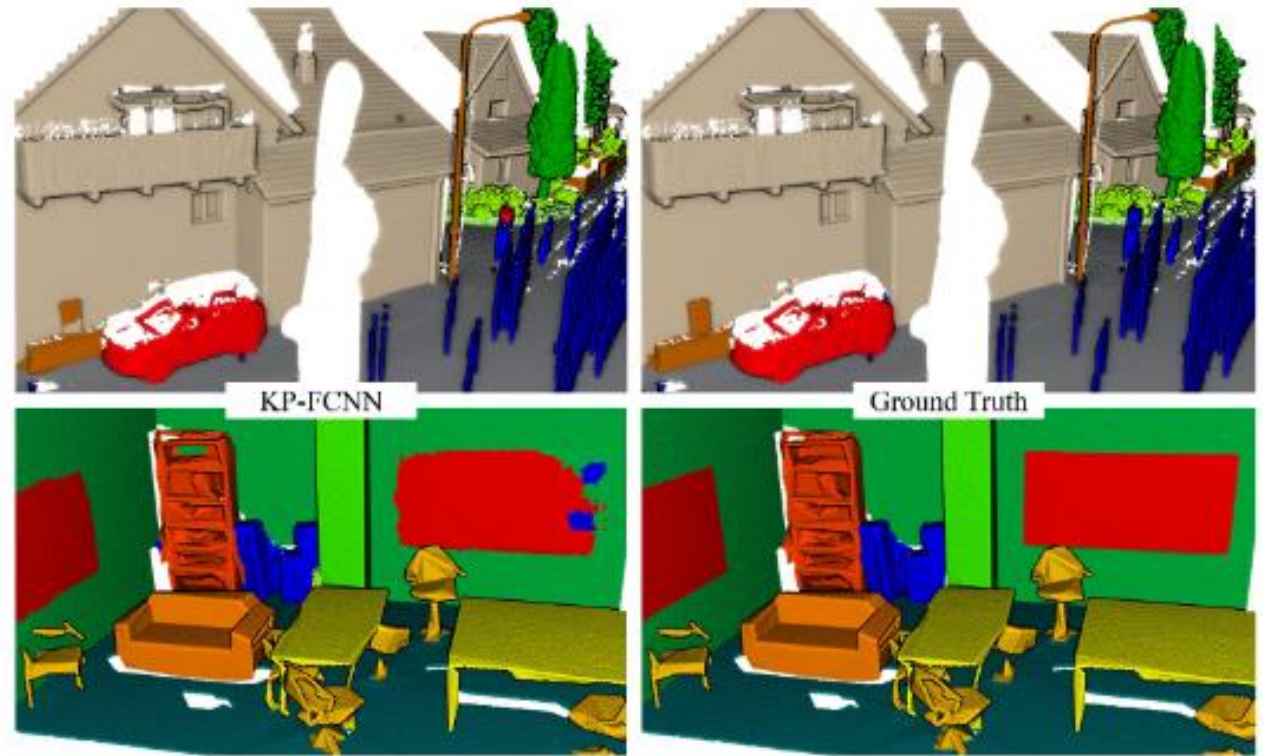
=> plus un point est loin d'un point du noyau, moins il s'exprimera à travers ce point



# Résultats

	ModelNet40	ShapeNetPart	
Methods	OA	mIoU	mIoU
SPLATNet [34]	-	83.7	85.4
SGPN [42]	-	82.8	85.8
3DmFV-Net [9]	91.6	81.0	84.3
SynSpecCNN [48]	-	82.0	84.7
RSNet [15]	-	81.4	84.9
SpecGCN [40]	91.5	-	85.4
PointNet++ [27]	90.7	81.9	85.1
SO-Net [19]	90.9	81.0	84.9
PCNN by Ext [2]	92.3	81.8	85.1
SpiderCNN [45]	90.5	82.4	85.3
MCCConv [13]	90.9	-	85.9
FlexConv [10]	90.2	84.7	85.0
PointCNN [20]	92.2	84.6	86.1
DGCNN [43]	92.2	85.0	84.7
SubSparseCNN [9]	-	83.3	86.0
KPConv <i>rigid</i>	<b>92.9</b>	85.0	86.2

Methods	Scannet	Sem3D	S3DIS	PL3D
Pointnet [26]	-	-	41.1	-
Pointnet++ [27]	33.9	-	-	-
SnapNet [4]	-	59.1	-	-
SPLATNet [34]	39.3	-	-	-
SegCloud [37]	-	61.3	48.9	-
RF_MSSF [38]	-	62.7	49.8	56.3
Eff3DConv [50]	-	-	51.8	-
TangentConv [36]	43.8	-	52.6	-
MSDVN [30]	-	65.3	54.7	66.9
RSNet [15]	-	-	56.5	-
FCPN [28]	44.7	-	-	-
PointCNN [20]	45.8	-	57.3	-
PCNN [2]	49.8	-	-	-
SPGraph [17]	-	73.2	58.0	-
ParamConv [41]	-	-	58.3	-
SubSparseCNN [9]	<b>72.5</b>	-	-	-
KPConv <i>rigid</i>	68.6	<b>74.6</b>	65.4	72.3



Réussir à capturer une forme géométrique locale ?

Fonction kernel  $g$  différentiable par la position des points du kernel

=> peut être appris

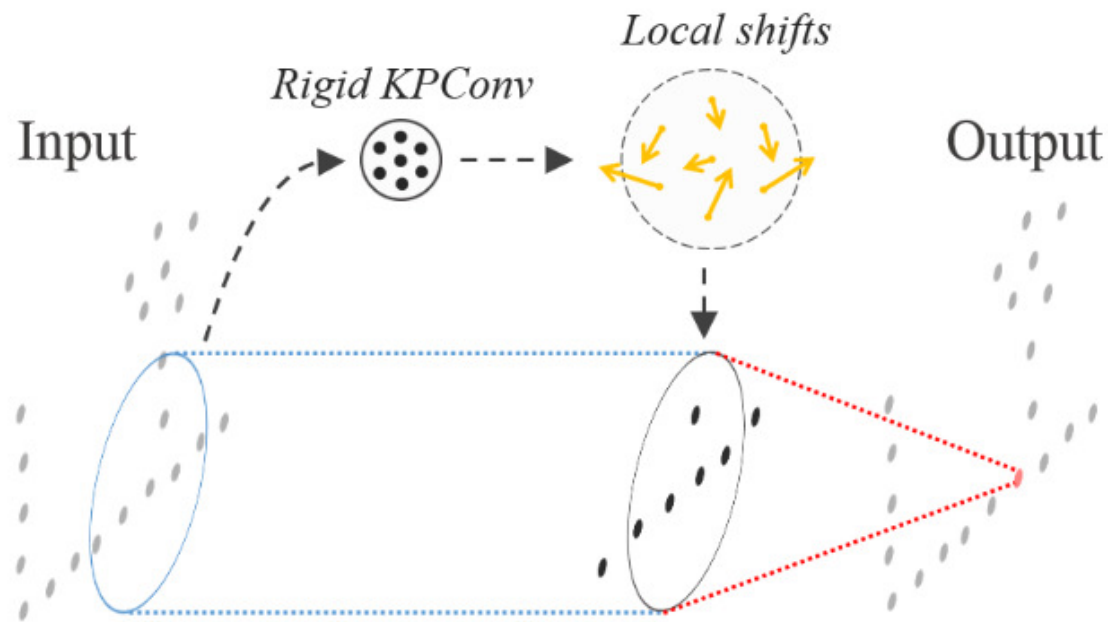
Similaire à l'approche 2D classique ("*Deformable convolutional networks*", Dai et al., 2017)

$$(\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g(x_i - x) f_i \quad \Rightarrow \quad (\mathcal{F} * g)(x) = \sum_{x_i \in \mathcal{N}_x} g_{\text{deform}}(x - x_i, \Delta(x)) f_i$$

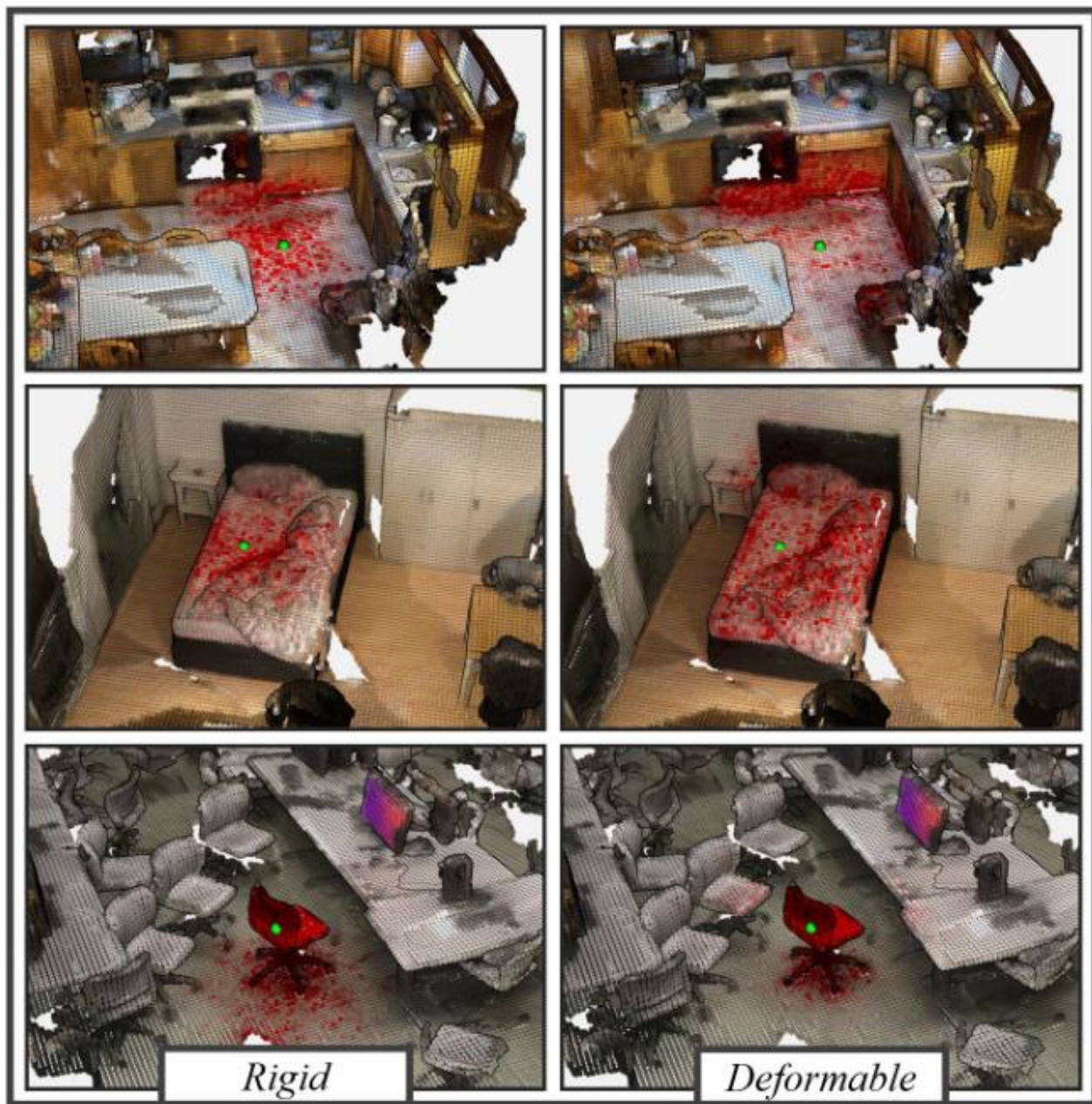
$$g(y_i) = \sum_{k < K} h(y_i, \tilde{x}_k) W_k \quad \Rightarrow \quad g_{\text{deform}}(y_i, \Delta(x)) = \sum_{k < K} h(y_i, \tilde{x}_k + \Delta_k(x)) W_k$$



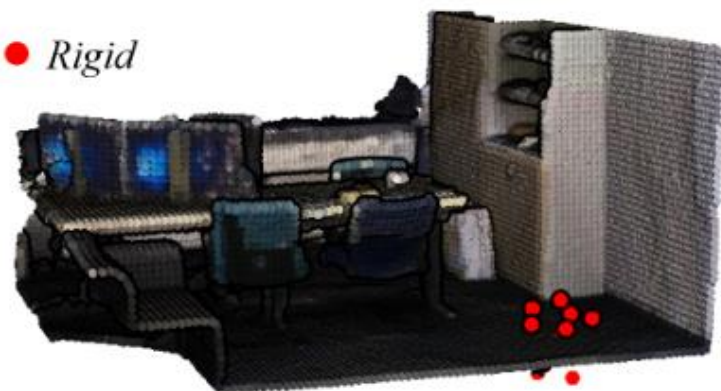
# Noyau de convolution **déformable** par point



- Réseau apprend deux éléments :
  - Shift à partir du noyau rigide
  - Extraction de features à partir de la convolution déformée
- Besoin de cadrer les décalages des points du noyau:
  - Pénaliser la distance entre un point du kernel et le point voisin le plus proche
  - Empêcher un trop gros overlap d'influence entre les points



● *Rigid*



● *No regularization*



● *With regularization*

