CAD-DEFORM: DEFORMABLE FITTING OF CAD MODELS TO 3D SCANS

Vladislav Ishimtsev*
Skolkovo Institute of Science and Technology
Russia

Alexey Bokhovkin*
Skolkovo Institute of Science and Technology
Russia

Alexey Artemov
Skolkovo Institute of Science and Technology
Russia

Savva Ignatyev
Skolkovo Institute of Science and Technology
Russia

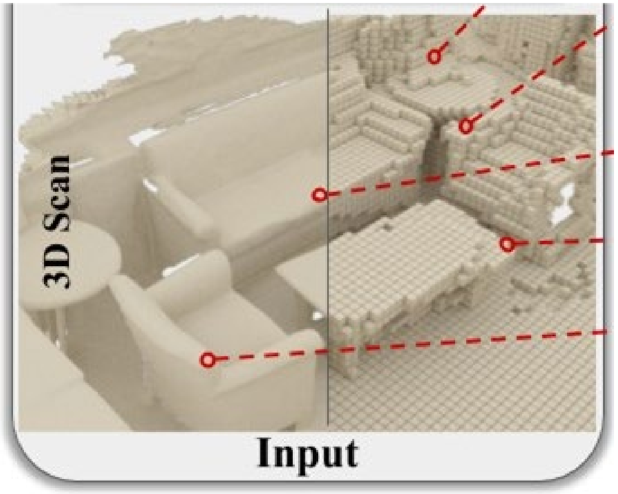
Matthias Niessner
Technical University of Munich, Germany

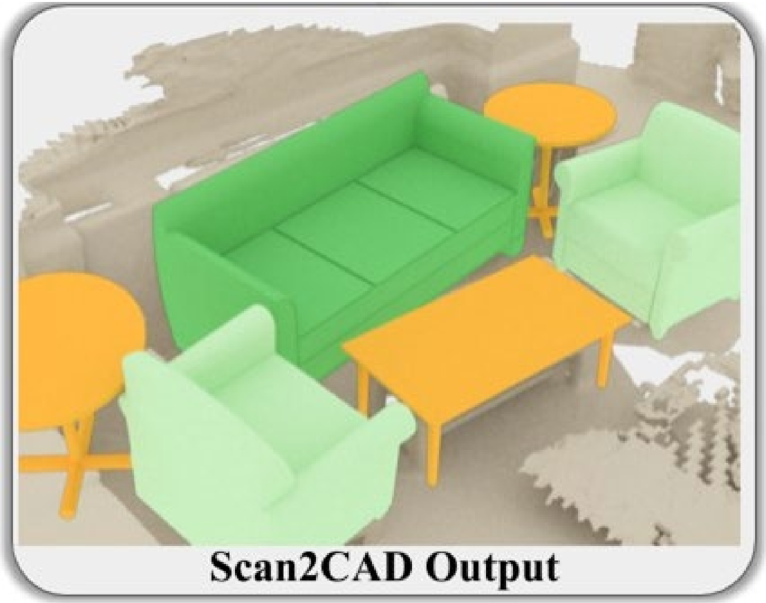
Denis Zorin
New York University, USA
Skolkovo Institute of Science and Technology
Russia

Evgeny Burnaev
Skolkovo Institute of Science and Technology
Russia

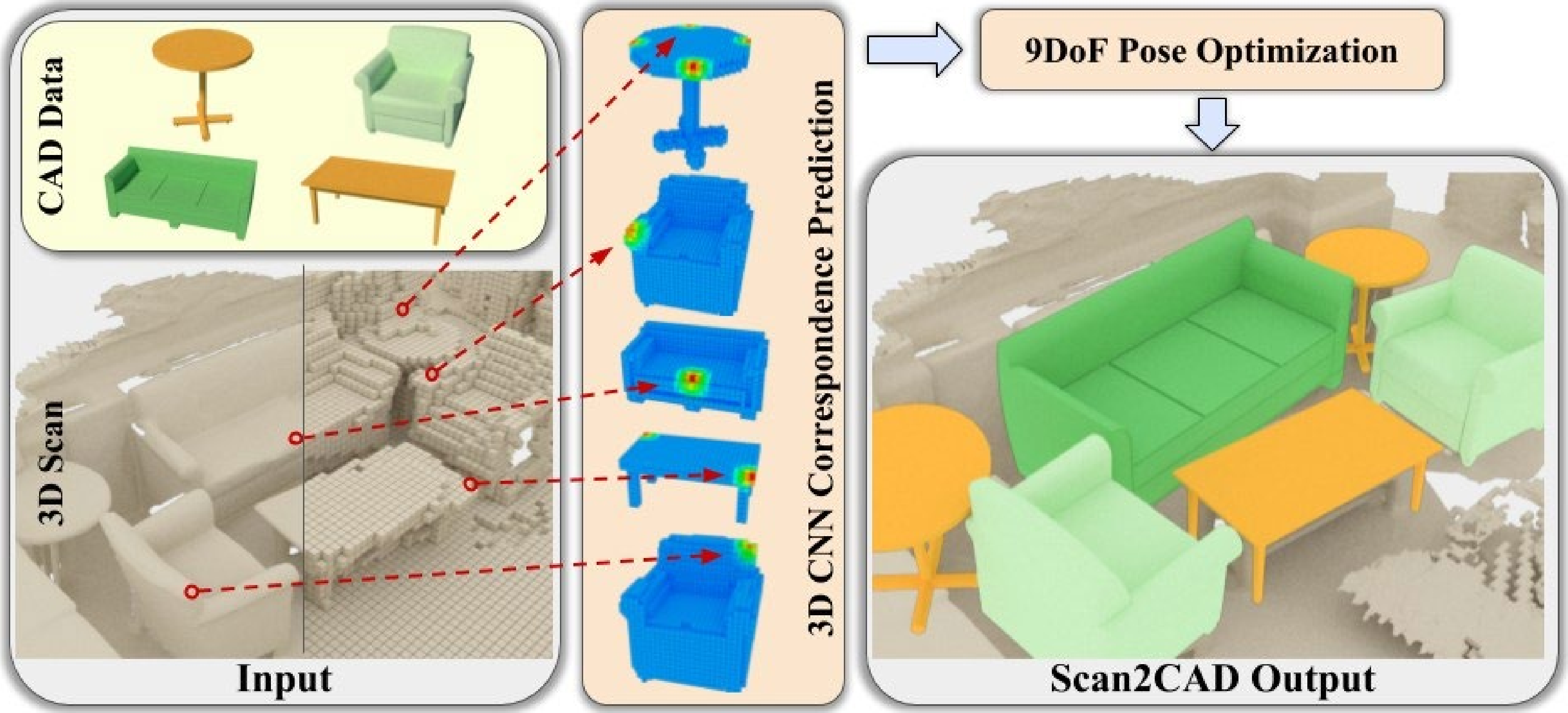
CS348N JC Kretsch 2022-02-28

The task: point cloud → set of meshes

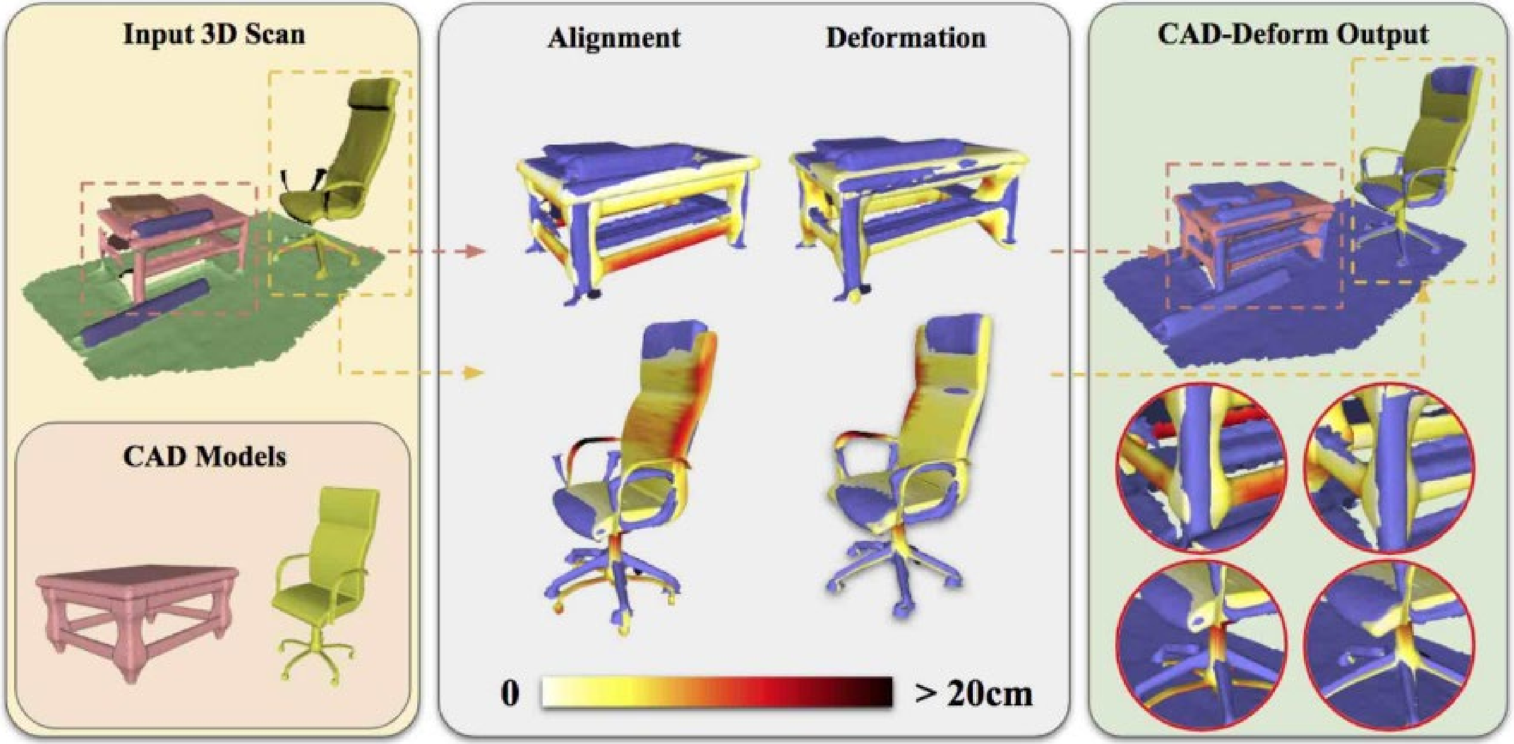




Past work: Scan2CAD (ScanNet+ShapeNet)

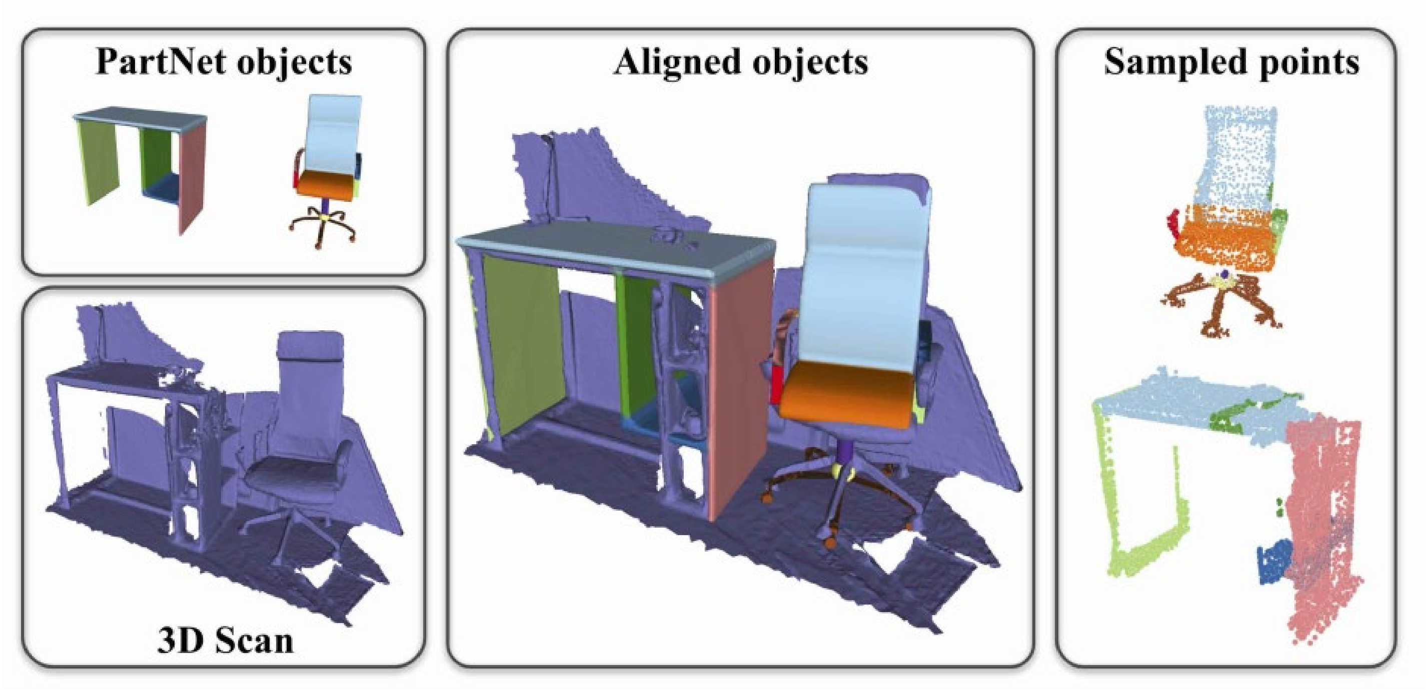


CAD-Deform



Input 3D Scan
CAD Models
Alignment
Deformation
0
> 20cm
CAD-Deform Output

Dataset - Scan2CAD (ScanNet+ShapeNet) +PartNet



Deformation: Optimize energy

Input:

$$\mathcal{E}(\mathbf{V},\mathbf{P}) = \underbrace{\boxed{E_{\text{shape}} + \alpha_{\text{smooth}} E_{\text{smooth}} + \alpha_{\text{sharp}} E_{\text{sharp}}}}_{\text{quadratic problem}} + \alpha_{\text{data}} E_{\text{data}},$$

$$E_{\text{shape}} = \underbrace{\sum_{e \in \mathbf{E}} \| T_e(\mathbf{V}) - T_e^0 \|_2^2;}_{\text{deviation}} \qquad E_{\text{smooth}} = \sum_{f \in \mathbf{F}} \sum_{e_i, e_j \in f} \| T_{e_i}(\mathbf{V}) - T_{e_j}(\mathbf{V}) \|_2^2;$$

$$E_{\text{sharp}} = \sum_{k=1}^{n_p} \sum_{e_s \in \mathbf{E}^k_{\text{sharp}}} \| T_{e_s}(\mathbf{V}) - T_{e_{s+1}}(\mathbf{V}) \|_2^2; \qquad E_{\text{data}} = f_{\text{data}}(\mathbf{V},\mathbf{P}).$$

How to compare vertices and points?

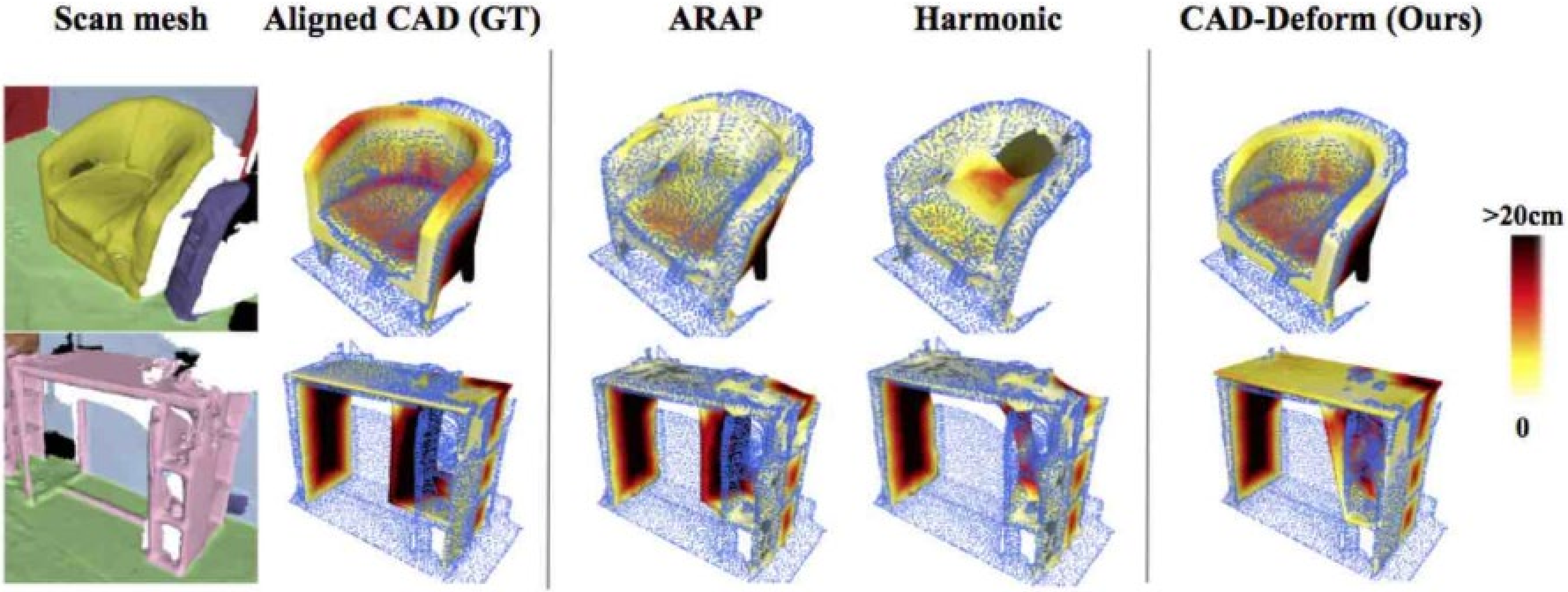
If start close to target, can use a priori correspondences:

$$f^{\text{nn}}_{\text{data}}(\mathbf{V},\mathbf{P}) = \sum_{c \in \mathbf{C}} \sum_{p \in \mathbf{P}_c} \| p - v_{\iota^c(p)} \|_2^2.$$

When far from target use screened attraction:

$$f^{\text{p2p}}_{\text{data}}(\mathbf{V},\mathbf{P}) = \sum_{c \in \mathbf{C}} \sum_{v \in \mathbf{V}_c} \sum_{p \in \mathbf{P}_c} \xi^{\sigma}(p, \mathbf{V}_c) \left(d^{\varepsilon}(v - p) \right)^2,$$

Results: deformations improve fit while maintaining shape



Results: deformations fit well independent of alignment input

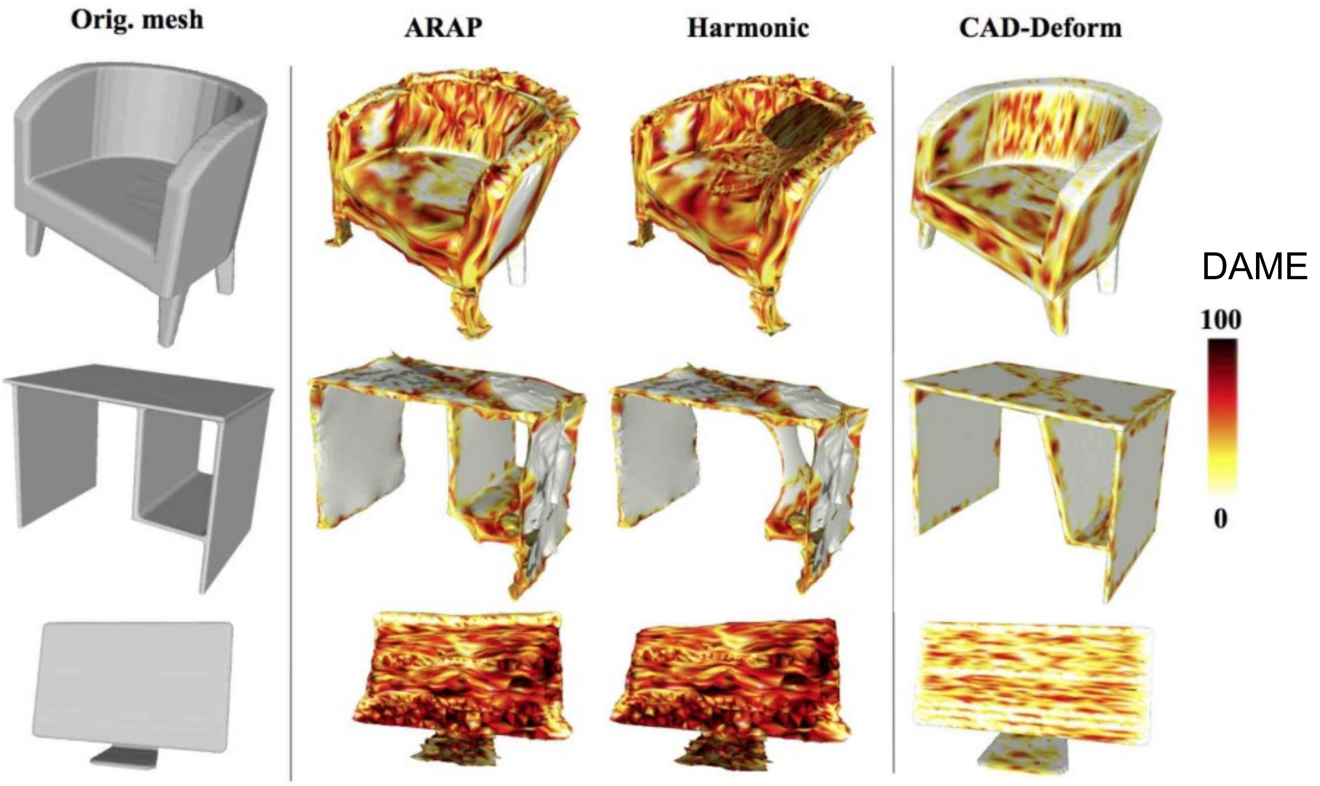
Method	Class avg. GT	Class avg. S2C [9]	Class avg. E2E [10]	Instance avg. GT	Instance avg. S2C [9]	Instance avg. E2E [10]
# TPs	1410	499	882	1410	499	882
TP undeformed	89.2	83.7	88.5	90.6	79.4	93.9
Ours: NN only	89.7	84.3	89.0	91.4	84.7	94.4
Ours: p2p only	90.3	88.3	89.4	91.6	90.3	94.9
Ours: w/o smooth	90.6	**90.0**	89.6	92.3	90.3	95.0
Ours: w/o sharp	90.3	86.9	**90.6**	92.3	89.4	**95.2**
CAD-Deform	**91.7**	89.8	90.3	**93.1**	**92.8**	94.6

Table 1: Comparative evaluation of our deformations to true positive (TP) alignments by non-deformable approaches in terms of Accuracy (%). Note that deformations improve performance for all considered alignment approaches.

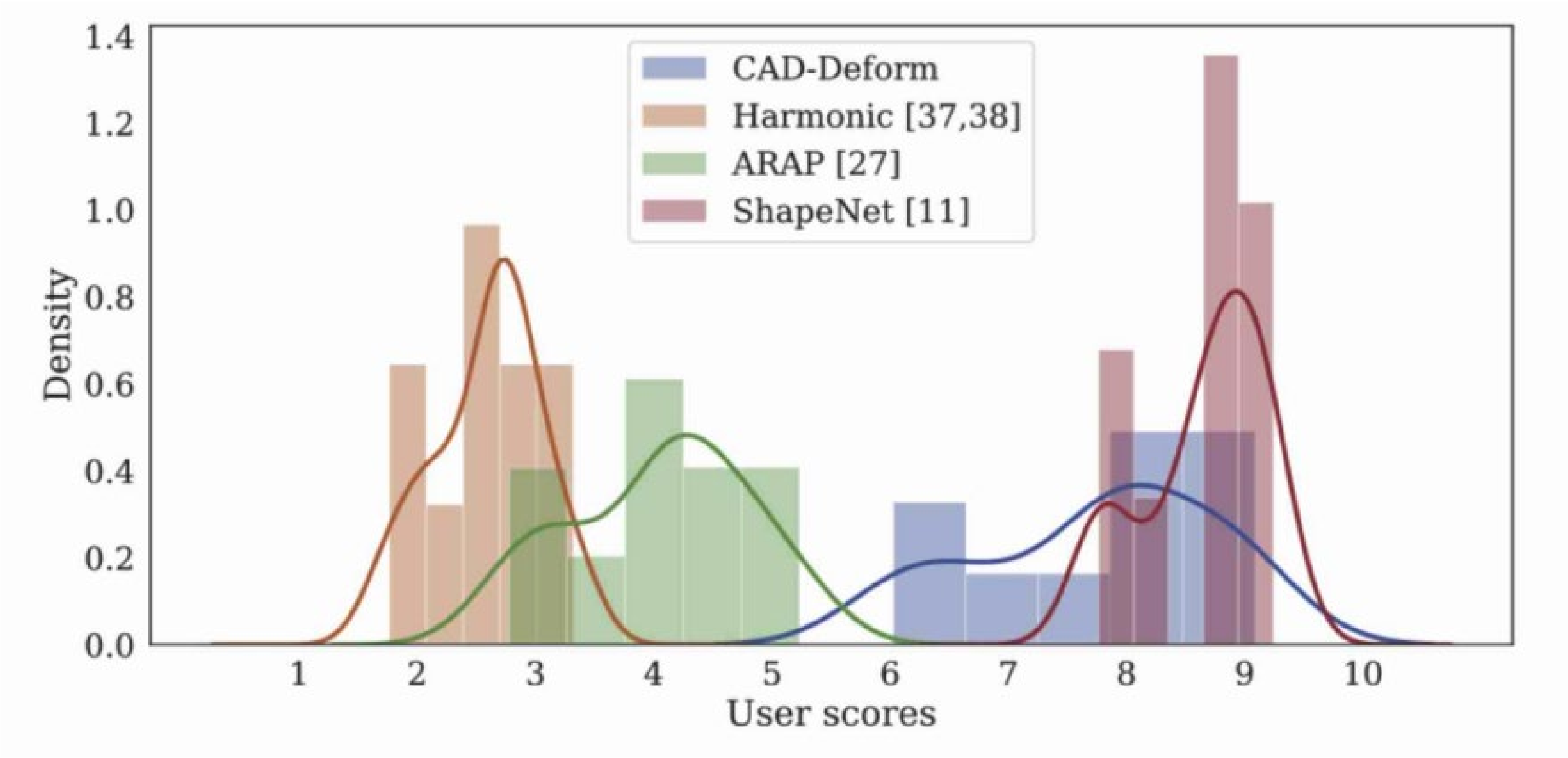
Method	bookshelf	cabinet	chair	display	table	trashbin	other	class avg.	avg.
# instances	142	162	322	86	332	169	197	201.4	1410
Ground-truth	88.0	75.2	94.8	98.9	89.6	96.6	81.4	89.2	90.6
Ours	**90.5**	**82.2**	**95.4**	**99.1**	**91.0**	**98.6**	**84.8**	**91.7**	**93.1**

Table 2: Comparative evaluation of our approach to non-deformable ground-truth and baselines in terms of scan approximation Accuracy (%). We conclude that our deformations improve fitting accuracy across all object classes by 2.5 % on average.

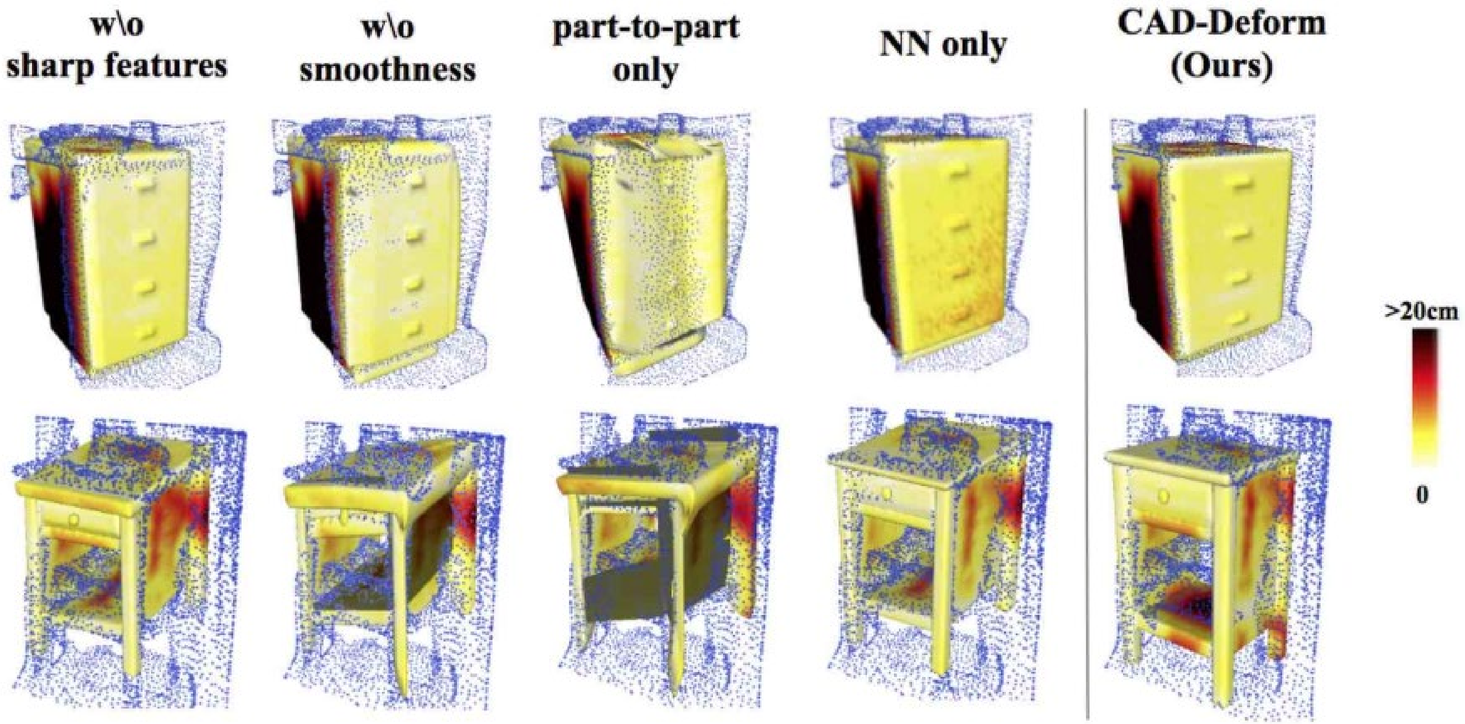
Results: produces higher quality surfaces



Results: produces higher quality surfaces



Results: balances fit-to-scan and perceptual quality

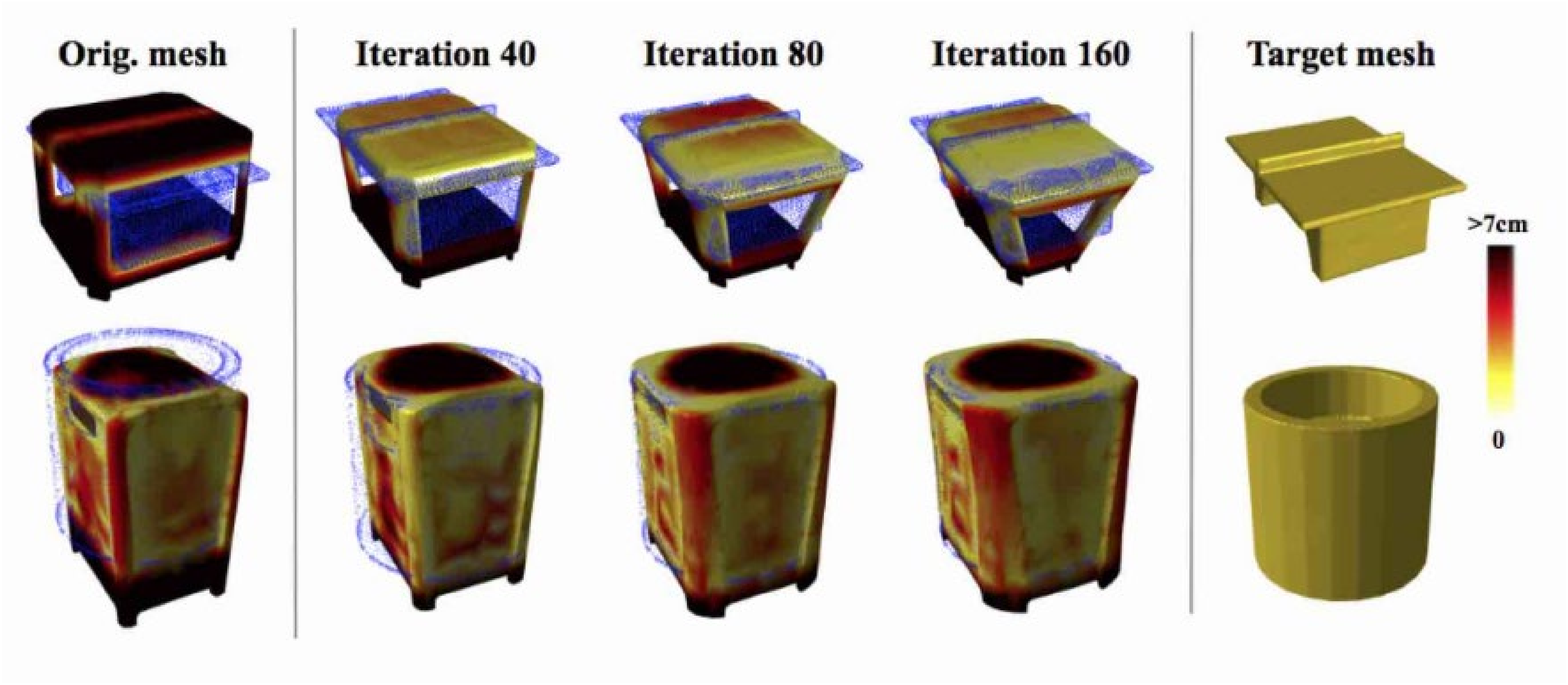


Conclusions

- Goal: mesh representations of real world scene from 3D scans
- Balance accuracy and high-perceptual quality
- Propose adding a deformation to Scan2CAD
- Deformation minimizes a composite energy function:
  - Uses semantic part structures
  - Enforces smooth transformations
  - Preserves sharp geometric features
  - Minimizes difference to point cloud
- First step in improving accuracy while preserving perceptual quality through deformation

Extra slides

Results: attempts shape interpolation



Results: attempts at various level of segmented objects

Accuracy, %	class avg.	avg.
Ground-truth	89.22	90.56
Level 1 (object)	89.25	90.79
Level 2	89.16	91.21
Level 3	89.40	91.05
Level 4 (parts)	**91.65**	**93.12**

Optimization of non-linear part

The data term is highly nonlinear, but solving the complete optimization problem can be done efficiently using A_{quad}^{-1} as the preconditioner. For our problem, we use the preconditioned L-BFGS optimizer summarized in Algorithm 1.

Algorithm 1: Preconditioned L-BFGS mesh optimization (PL-BFGS)

$M_{\text{precond}} = A_{\text{quad}}^{-1}$ `// stored as LU decomposition`

$\overline{\mathbf{V}} = T_m^0(\overline{\mathbf{V}}^0)$

for $i \leftarrow 0$ **to** N_{iter} **do**

$\quad g_{\text{tot}} = \alpha_{\text{data}} {}^{\mathrm{d}E_{\text{data}}}/_{\mathrm{d}p} + A_{\text{quad}}\overline{\mathbf{V}} + b$

$\quad \overline{\mathbf{V}} = \text{L-BFGS-step}(\overline{\mathbf{V}}, g_{\text{tot}}, M_{\text{precond}})$

end