IGE-Net: Inverse Graphics Energy Networks for Human Pose Estimation and Single-View Reconstruction

Dominic Jack[1] d1.jack@qut.edu.au
Frederic Maire[1] f.maire@qut.edu.au
Sareh Shirazi[1] s.shirazi@qut.edu.au
Anders Eriksson[2] a.eriksson@uq.edu.au

[1]School of Electrical Engineering and Computer Science, Queensland University of Technology
[2]School of Information Technology and Electrical Engineering, University of Queensland

Abstract

Inferring 3D scene information from 2D observations is an open problem in computer vision. We propose using a deep-learning based energy minimization framework to learn a consistency measure between 2D observations and a proposed world model, and demonstrate that this framework can be trained end-to-end to produce consistent and realistic inferences. We evaluate the framework on human pose estimation and voxel-based object reconstruction benchmarks and show competitive results can be achieved with relatively shallow networks with drastically fewer learned parameters and floating point operations than conventional deep-learning approaches.

1. Introduction

Computer graphics involves reducing 3D scene information to 2D using well-understood physics-based arguments and mathematical operations like frame transformations and projections. Computer vision can be thought of as the inverse problem – inferring 3D scene information from some 2D representation.

Unlike graphics, computer vision is inherently ill-posed. While it is straight-forward enough to obtain an inference which is consistent with a given 2D representation using standard graphics and optimization techniques, there is no guarantee this inference will be realistic. To resolve this, we propose using simple optimization techniques on a learned energy function which combines graphics operations with a learned realism component.

We learn this energy function itself using deep-learning optimization techniques, resulting in a multi-level optimization framework which can be trained end-to-end. We apply our framework to two common problems: 3D human pose estimation, and single-view voxel-based object reconstruction.

2. Main Contributions

Our main contributions are as follows.

1. We propose simple parameterized energy functions that capture both consistency and feasibility for the problems of human pose estimation and object reconstruction based on 2D features and well-understood computer-graphics principles.

2. For the case of human pose estimation, we show the proposed energy function can be used to lift 2D pose inferences to 3D at competitive accuracies with significantly fewer learned parameters and computational requirements.

3. For object reconstruction, we demonstrate the framework can produce high-resolution voxel grids from single images on standard desktop GPUs without the need for 3D convolutions or deconvolutions, outperforming state-of-the-art high-resolution methods in terms of accuracy.

3. Prior Work

3.1. Multi-Level Optimization

Many problems in machine learning involve inferring values of unknown variables from observations. Energy-based models describe relationships between sets of variables by mapping each combination to a scalar energy value, where realistic combinations correspond to lower energies than their less viable counterparts. Inferences are made by fixing values of known variables and seeking unknown values which minimize the energy [27].

Energy-based models have been combined with deep learning in the past. Zheng *et al*. [59] formulated conditional random fields (CRFs) as a recurrent neural network layer, which combined with a standard convolutional neural network (CNN) achieved state-of-the-art results for image

segmentation. Amos and Kolter [1] considered energy functions based on quadratic programs. Their implementation solved the inner optimization problem efficiently and exactly, and demonstrated it was able to learn hard constraints like those associated with the number-game Sudoku.

Domke [13] presented a number of implementations for efficiently computing and differentiating approximate optimizations – solutions where the energy minimization process is based on a fixed number of steps of some optimization algorithm. While the algorithms did not find the exact solution to the energy minimization problem, these truncated optimization processes still yielded good results for image denoising and labeling problems. Belanger *et al*. [3] took a similar approach and showed inexact optimization of complex energy functions outperformed exact solutions using simpler functions for image denoising and natural language semantic role labeling.

3.2. Human Pose Estimation

Inferring human pose in two or three dimensions from images is an important part of many tasks including human-computer interaction and action recognition. For the 2D problem, traditional approaches combine visual features and image descriptors with a tree-structure of the body and known invariants and proportions [58]. More recently, deep learning's wave of success in other image processing applications such as image classification and segmentation has flowed into pose estimation, with fully-convolutional approaches achieving exceptionally accurate 2D inferences by regressing heatmaps rather than the joint coordinates themselves [50, 35, 10, 5, 11].

The 3D problem is considerably more challenging. In addition to problems involved in the 2D variant, the main difficulty in training 3D pose inference systems that work in the wild is the availability of varied datasets. While 2D datasets can be annotated manually, 3D information is generally gathered using special motion-capture systems. Although these systems are capable of generating massive volumes of data, the examples within such datasets are usually limited in variety. For example, the human 3.6 million dataset (H3M) [19] contains millions of frames, but all images are collected in the same room with only a handful of subjects. By contrast, the popular 2D dataset COCO [30] features over 50,000 human pose annotations with very few duplicates.

To get around this lack of varied 3D data, many methods use a 2-stage approach to 3D inference by inferring 2D poses from images, then lifting these 2D poses to 3D separately [4, 7, 33]. These approaches benefit from the varied image features in 2D datasets, but the separate stages means any "lifting" module is unable to take advantage of contextual information learned in the first stage.

The other main difficulty with 3D pose estimation is the inherent ambiguity associated with depth inference and occlusions. Adversarial approaches tackle this by introducing loss terms which are themselves learned in a modified minimax game [21, 47, 56].

3.3. Single View 3D Object Reconstruction

Reconstructing 3D objects from a single view is a common problem in computer vision and robotics. Fundamental to any approach is the representation of the output object. Volumetric methods are the most wildely used in 3D learning [48, 54, 8, 9, 20, 55, 37, 51, 24, 60]. These approaches generally use 3D analogues of ideas and operations that have proven successful in image processing, including convolutions, deconvolutions and feature pooling. Recent advances in auto-encoders [15, 43] and GANs [53, 52, 31, 17] have also shown promising results on regular 3D grids, while Tulsiani *et al*. [46] showed object shape and pose can be learned simultaneously and without 3D labels using only depth maps or silhouettes to encourage view consistency across multiple views.

Unfortunately, the additional dimension inherent to 3D representations means these methods scale poorly with resolution, resulting in generally coarse outputs – typically 32^3 or 64^3. To overcome this scaling issue, octree networks [40, 49, 18, 45] recursively divide regions of interest into octants. By focusing only on regions near the object surface, these methods operate with complexity proportional to surface area rather than volume.

Other approaches to high-resolution inference keep the regular volumetric data structure but use operations that scale better to higher resolutions [23, 39].

Point cloud methods avoid the need to discretize space, instead working on continuous coordinates of points on the object surface [14, 36, 38, 28]. However, the variable size and unordered nature of point clouds introduce their own complexity in deep learning frameworks. Template deformation approaches [26, 22, 57] instead infer a constant-sized space warping that can be applied to an arbitrarily dense cloud or mesh. This comes at a cost however, as the topology of the output shape is intrinsically coupled with that of the deformed template.

4. Method Overview

Our approach is based on energy minimization networks which have been discussed previously in the literature [27, 13, 2, 3]. We base our notation on the work of Belanger, McCallum and Yang [3], where we seek the minimizer of some energy function

$$\arg\min_{\tilde{\mathbf{y}}} E(\tilde{\mathbf{y}}; \mathbf{x}, \theta_E). \tag{1}$$

We implement the energy function E as a neural network which takes as inputs a proposed solution $\tilde{\mathbf{y}}$ and extracted

features $\mathbf{x}$ with learned parameters θ_E. For generic non-convex energies, calculating the exact arg min is intractible, hence we approximate the result by the output of some iterative strategy

$$\tilde{\mathbf{y}}^{(t)} = \mathbf{f}(\tilde{\mathbf{y}}^{(t)}, E(\tilde{\mathbf{y}}^{(t-1)}; \mathbf{x}, \theta_E); \theta_{\text{opt}}), \tag{2}$$

for some fixed number of steps $t \in [1, T]$, where θ_{opt} are hyper-parameters of the optimization strategy and $\tilde{\mathbf{y}}^{(0)}$ is an initial proposal. For example, basic gradient-descent with learning rate η is implemented as

$$\mathbf{f}(\tilde{\mathbf{y}}, E(\tilde{\mathbf{y}}; \mathbf{x}, \theta_E); \eta) = \tilde{\mathbf{y}} - \eta \nabla_{\tilde{\mathbf{y}}} E. \tag{3}$$

In this investigation we also considered gradient-descent with momentum and gradient-clipping, where the momentum term and clip value were trained as part of θ_{opt}.

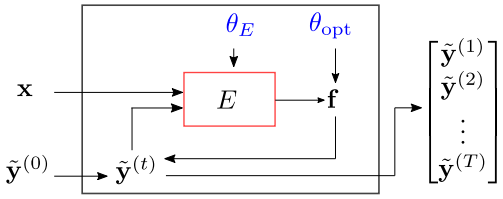

Figure 1: Unrolled optimization involves iteratively updating a proposed value $\tilde{\mathbf{y}}^{(t)}$ to minimize some energy function E according to an update step $\mathbf{f}$. Parameters of E and $\mathbf{f}$ (blue) are learned in the outer optimization process.

This process is illustrated in Figure 1. We refer to this scheme as *unrolled gradient descent* or *inner optimization*.

To train our network we use a loss λ made up of a weighted sum of losses applied to all steps of the optimization process,

$$\lambda = \sum_{t=0}^{T} \hat{\lambda}(\tilde{\mathbf{y}}^{(t)}, \mathbf{y}). \tag{4}$$

where k_t is a scalar weighting value, $\mathbf{y}$ is the example label and $\hat{\lambda}$ is some per-proposal loss function dependent on the problem. In all experiments we use exponential weighting $k_t = 0.9^{T-t}$.

Assuming E and $\mathbf{f}$ are piecewise-doubly-differentiable and λ_0 is piecewise differentiable, the parameters θ_E and θ_{opt} can be learned using any standard optimization strategy referred to as the *outer optimizer*. For brevity, we drop the parameters θ_E and θ_{opt} in equations and diagrams hereafter.

To summarise, our inverse graphics energy networks (IGE-Net) are made up of:

- a feature extractor module that provides a (possibly empty) set of features as well as an initial estimate;
- an energy module which reduces a proposed solution and observed features to a scalar value; and
- an inner optimization strategy.

5. Human Pose Estimation

We begin by considering the problem of lifting human joint information from 2D ($\mathbf{x} \in \mathbb{R}^{N_{J_2} \times 2}$) to 3D ($\mathbf{y} \in \mathbb{R}^{N_{J_3} \times 3}$). Note we do not require the number of joints to be the same, nor do we require any known correspondences between the two sets. This allows us to pair 2D inferences from a model trained on one dataset with 3D pose data with different joint annotations.

Recent progress in this area has resulted in a number of algorithms performing very well on standard benchmarks, split on accuracy metrics by a matter of millimeters. For many applications, such error rates are well and truly satisfactory, so we approach this problem with the aim to minimize memory requirements and computational costs – factors more important in areas such as mobile robotics and autonomous systems – while maintaining reasonable accuracy.

We also limit our methods to perform well as defined by scale-invariant metrics. While scale can be learned based on contextual information, errors in scale inference tend to drown-out those associated with relative positions.

5.1. Network Structure

We base our feature extractor module on the work of Martinez *et al.* [33]. The proposed network is made up of two residual blocks each containing two dense layers along with an input and output layer for a total of six, as well as batch normalization, rectified linear activations, weight clipping, residual connections and dropout. While this network is small by modern standards, we reduce it further by removing one of the internal blocks and dropping the number of units in each remaining inner layer by a factor of 8. This reduces the number of trainable parameters by roughly a factor of 100. Since our losses and evaluation are scale-agnostic, we also remove the weight clipping.

We consider an energy function E as the combination of a *reprojection energy* $E_{\mathbf{x}}$ and a *feasibility energy* $E_{\mathbf{y}}$,

$$E(\tilde{\mathbf{y}}; \mathbf{x}) = E_{\mathbf{x}}(\tilde{\mathbf{x}}(\tilde{\mathbf{y}}); \mathbf{x}) + E_{\mathbf{y}}(\tilde{\mathbf{y}}), \tag{5}$$

where $\tilde{\mathbf{x}}(\tilde{\mathbf{y}})$ is the projection of the proposed solution. We assume the intrinsic camera parameters are known, and infer 3D poses in the camera's reference frame.

Each energy function makes use of pairwise squared euclidean distances similar to Moreno-Noguer [34],

$$\Delta^2_{ij}(\mathbf{z}) = ||\mathbf{z}_i - \mathbf{z}_j||^2_2, \; j > i, \tag{6}$$

where $\mathbf{z}$ is an ordered set of points in $\mathbb{R}^N$. This transformation has many desirable properties, including invariance to rotation, translation and reflection. Unlike Moreno-Noguer, we use the squared distance rather than the actual difference, as this avoids a square root operation causes problems with gradients near zero.

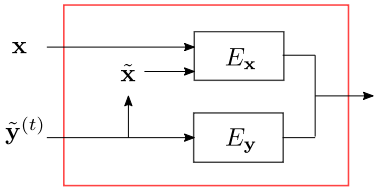

Figure 2: For lifting 2D pose information to 3D, we split the energy into 2 parts: a reprojection loss $E_{\mathbf{x}}$ which measures how consistent the projected proposed pose is with the observed 2D information, and a feasibility loss $E_{\mathbf{y}}$ which operates on the normalized proposed pose.

We parameterize our reprojection loss as a 2-layer dense network $\mathrm{DN}_{\mathbf{x}}$ with a softplus and softabs activation. Inputs are given by the pairwise squared distances between all points in $\mathbf{x}$ and $\tilde{\mathbf{x}}$, i.e.

$$E_{\mathbf{x}}(\tilde{\mathbf{x}};\mathbf{x}) = \mathrm{DN}_{\mathbf{x}}\left(\Delta^2(\tilde{\mathbf{x}} \oplus \mathbf{x})\right), \tag{7}$$

where $\oplus$ is the concatenation operator along the joint dimension.

While a perfect proposal will yield a perfect reprojection ($\tilde{\mathbf{y}} = \mathbf{y} \Rightarrow \tilde{\mathbf{x}} = \mathbf{x}$), the reverse implication does not hold. As the name suggests, the feasibility energy $E_{\mathbf{y}}$ is intended to promote feasible proposals independently of the appearance $\tilde{\mathbf{x}}$. To make this scale-invariant, we normalize the proposed pose $\hat{\mathbf{y}} = N(\tilde{\mathbf{y}})$ by dividing by the distance between the hip joints, then consider the pairwise squared distances,

$$E_{\mathbf{y}}(\tilde{\mathbf{y}}) = \mathrm{DN}_{\mathbf{y}}(\Delta^2(\hat{\mathbf{y}})). \tag{8}$$

This energy architecture is illustrated in Figure 2.

To train our model, we use a per-step outer loss function

$$\hat{\lambda}(\tilde{\mathbf{y}}^{(t)}, \mathbf{y}) = ||k\tilde{\mathbf{y}}^{(t)} - \mathbf{y}||_2, \tag{9}$$

where k is the optimal scaling factor with respect to the squared error.

5.2. Implementation Details

We pretrain our initial pose estimation network independently for 200 epochs with a batch size of 64 as per the original [33].

For our inner loss networks, we initialized the hidden layer weights using Glorot initialization [16], and the loss layer weights with a version scaled down by 10^{-3}. This resulted in the inner optimizer starting with little effect and growing, which smooths learning in the very early stages. We used the same learning-rate decay schedule as Martinez *et al*. [33] except with initial learning rate lowered by a factor of 10 and training to convergence.

We initialize our inner optimizer's learning rate, gradient clip value and momentum at 1, 1 and 0.1 respectively. To prevent negative momentum early due to spurious gradients in the initial loss function, we used the absolute value of a learned parameter rather than the learned parameter itself.

We run experiments on the popular Human 3.6 million (H3M) dataset [19]. We use 2D pose inferences provided by Martinez *et al*. [33] which come from stacked hourglass networks of Newell *et al*. [35]: one trained entirely on varied 2D poses in-the-wild, and another fine tuned on H3M. We also experiment with ground-truth 2D poses. All training and evaluation uses inputs with the 16 joints used in COCO [30], and infer a slightly different set of 16 joints in 3D. Evaluation is on a 17-joint skeleton with additional pelvis joint as per Martinez *et al*. [33]. We train on subjects 1, 5, 6, 7 and 8 and evaluate on subjects 9 and 11.

5.3. Results

Sample results for our 2-step network trained on 2D ground-truth inputs are shown in Figure 3. We see the network learns to reconcile inconsistent 2D data in a single step. The subsequent step has a smaller impact, but still makes minor adjustments to the 3D pose without losing consistency with the observation.

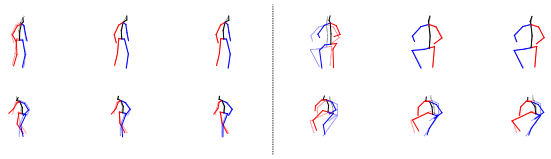

Figure 3: Camera view (top) and novel view (bottom) of inferred pose (solid) and ground truth (dotted) after 0, 1 and 2 steps. Note the observed 2D pose (dotted, top) has one fewer joints in the head. The model uses camera view 2D joint coordinates (top, dotted) as inputs.

We evaluate our models using two metrics: the mean per-joint error after scaling as per Equation 9, and the per-joint error after an optimal rigid body transformation. We refer to these as Protocol 1a and Protocol 2 respectively (Martinez *et al*. [33] define Protocol 1 to be a slightly different metric. It is largely analogous in meaning to our Protocol 1a, though not equivalent).

We begin analysis by looking at performance of our networks using ground truth 2D poses with different numbers of inner optimizer steps. We compare against different versions of the base model without the IGE component by varying the number of residual blocks as well as the number of hidden units in each layer. Protocol 2 results are shown in Figure 4.

Our IGE networks can achieve competitive results in a handful of steps, with performance comparable to the full base model with significantly few operations. Unlike the baseline, our networks also have a constant number of

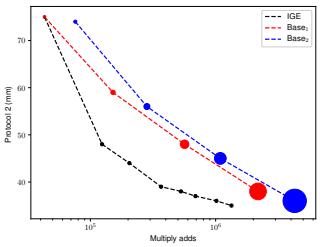

Figure 4: Protocol 2 scores (lower is better) and the number of multiply-adds due to dense layers in inference. Base model values are for networks with (left-to-right) 128, 256, 512 and 1024 units in each dense layer and 1 (red) or 2 (blue) residual blocks. IGE values (black) are for (left-to-right) 0, 1, 2, 4, 6 and 8, 12 and 16 steps. The size of each dot represents the number of trainable parameters of the model.

	Protocol 1a			Protocol 2		
2D source	SH	FT	GT	SH	FT	GT
Mart. [33]	-	-	-	52.5	47.7	37.1
Base 1024/2	79.0	75.1	61.6	52.2	47.9	35.8
IGE_4	75.1	67.8	45.1	56.1	51.5	39.4
IGE_8	72.8	66.0	42.6	55.1	50.5	37.7

Table 1: Average Protocol 1a/Protocol 2 scores for inferences based on stacked hourglass detections (SH), fine-tuned stacked hourglass detections (FT) and ground truth projections (GT). Baseline models had 1024 hidden units and 2 residual blocks. IGE networks were trained for 4 and 8 steps. Lower is better.

trained parameters, resulting in a significantly smaller memory footprint.

Results for experiments based on inferred 2D poses are shown in Table 1. Interestingly, our baseline method appears to overfit certain displacements, resulting in a relatively high Protocol 1a loss, though achieves a loss consistent with Martinetz *et al*. after optimal translation. Our IGE network performs slightly worse than Martinez *et al*. on inferred detections, though given the reduced computational and memory burden we believe this would be an acceptable trade-off in many scenarios.

6. Single View 3D Object Reconstruction

For the problem of 3D object reconstruction we parameterize shapes as voxel occupancy grids and seek a method that will scale well to high resolutions.

6.1. Energy Formulation

Theoretically, the approach of separating reprojection and feasibility losses can be applied to object reconstruction by comparing silhouettes and using some 3D convolutional encoder respectively. However, initial experiments showed this approach suffered from a number issues. These included issues with formulating projections of continuous-valued proposed solutions and scaling issues associated with the cubic nature of the grid.

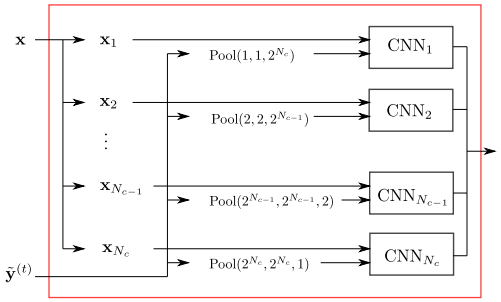

Figure 5: Energy function for single view reconstruction.

Instead, we propose a very different energy function formulation for single view reconstruction. We consider the inner optimizer input $\mathbf{x}$ to be the progressive outputs of some 2D convolutional network with N_C output feature-banks of different resolutions $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_C}\}$.

We consider an energy function made up of the sum of energy functions at each resolution. For each image feature map $\mathbf{x}_i$ of shape (h_i, w_i, f_i) we consider a voxel grid in the camera's viewing frustum $\bar{\mathbf{y}}_i$ of shape (h_i, w_i, d_i) by averaging the proposed voxel grid values in world coordinates $\tilde{\mathbf{y}}$ over the frustum voxel volumes. Our energy function seeks to learn the consistency between all voxels values along a ray and the image features of the associated pixel,

$$E(\tilde{\mathbf{y}}; \mathbf{x}) = \sum_i \text{CNN}_i(\mathbf{x}_i \oplus \bar{\mathbf{y}}_i), \tag{10}$$

where concatenation is along the feature dimension and CNN_i is some short 2D convolutional neural network.

By setting the depth of the averaged frustum voxel grid d_i and the number of filters in each layer of CNN_i to be proportional to the number of image features f_i, and assuming those image features roughly double in depth as they halve in spatial resolution, we ensure the number of operations at each image resolution is the same. This allows for much better resolution scaling than typical 3D convolution/deconvolution networks.

In practice, averaging a voxel grid in world coordinates over voxels corresponding to a frustum grid is a non-trivial operation and must be done at each step and resolution of the inner optimizer across all examples. Instead, we transform the labels of our dataset into the frustum space in a preprocessing step. During inference, the proposed solution $\tilde{\mathbf{y}}$ is a voxel grid in frustum coordinates which is average pooled anisotropically with different pool sizes for each image resolution. This means only the average pool-

ing must be done at each inner optimization step and resolution. Although this pooling operation still scales proportionally to the number of voxels and inner optimization steps ($\mathcal{O}(TN^3)$), GPU pooling implementations are relatively fast and the operation introduces no additional parameters.

While this means our method requires knowledge of the intrinsic camera parameters, we argue the choice of frame is arbitrary. Our method does not explicitly use the pose of the camera in its inference, and while the dataset transformation discussed above results in a slightly different problem compared to other approaches in the literature, we do not believe this puts us at an unfair advantage. On the contrary, the transformation results in a more varied dataset, and we demonstrate experimentally that traditional approaches perform slightly worse in this environment.

Our energy architecture is illustrated in Figure 5.

6.2. Outer Loss

For training, we experimented with two different per-step outer losses. Firstly, we consider an α-balanced focal loss [29] based on cross-entropy,

$$\hat{\lambda}_{CE}(\tilde{\mathbf{y}},\mathbf{y}) = -\sum_v [y_v(1-\tilde{y}_v)^\gamma(1+\alpha)\log(\tilde{y}_v) + (1-y_v)\tilde{y}_v^\gamma(1-\alpha)\log(1-\tilde{y}_v)], \tag{11}$$

where summation is over all voxels v. This is a generalization of standard cross-entropy (which is recovered by setting $\gamma = \alpha = 0$) designed to alleviate issues with class imbalance. $\alpha \in (0,1)$ results in additional focus on positive examples, while $\gamma > 0$ results in reduced focus on easy examples like those associated with the outside (usually empty) or very center (usually filled) of the voxel grid.

Secondly, we experiment with a continuous intersection-over-union implementation similar to that proposed by Richter and Roth [39],

$$\hat{\lambda}_{IoU}(\tilde{\mathbf{y}},\mathbf{y}) = 1 - \frac{\tilde{\mathbf{y}}\cdot\mathbf{y}}{||\tilde{\mathbf{y}}+\mathbf{y}||_1 - \tilde{\mathbf{y}}\cdot\mathbf{y}}. \tag{12}$$

6.3. Implementation Details

We experimented with two architectures: a small network with encoder based on MobilenetV2 (MN) [42], and another larger network based on Inception-V4 (I4) [44].

Image decoding networks built off the encoder network following a typical U-Net architecture common in the literature [41, 32, 35]. For the initial estimate, we used the output of a 3D deconvolution network based on the generator of Wu *et al*. [53] with one fewer layers, producing an output of resolution 32^3. We then trilinearly upsampled to the required resolution.

The inner-loop CNNs (CNN_i) each consist of two 3×3 2D convolutions without padding except the lowest resolution, which was a 3×3 followed by a 2×2, with softplus and softabs activations.

Our inner optimizer used a learned learning rate and gradient clip value. We observed no significant difference with momentum, so did not include it in experiments.

We used a base-line 3D deconvolutional network for low-resolution comparison (32^3) similar to the initial estimate network, except we doubled the number of features to keep the number of trained parameters comparable.

An overview of feature sizes and parameter counts is given in Table 2. Additional details and network diagrams are provided in the supplementary material.

		Base		IGE	
		MN2	I4	MN2	I4
Image Encoder	Output size	1280	1536	$4^2\times320$	$4^2\times1536$
	Parameters	2,223,872	54,276,192	1,811,712	54,276,192
3D Decoder	Initial size	$4^3\times128$	$4^3\times512$	$4^3\times64$	$4^3\times256$
	Parameters	2,656,113	14,159,297	238,009	3,802,849
Image Decoder	Initial size	-	-	$4^2\times128$	$4^2\times512$
	Parameters	-	-	140,992	2,928,384
Inner-loop CNN	Initial size	-	-	$4^2\times256$	$4^2\times1024$
	Parameters	-	-	1,109,840	16,573,760
Inner Optimizer	Parameters	-	-	2	2
Total	Parameters	4,879,985	68,435,489	3,300,555	77,581,187

Table 2: Network specification summary. Parameter counts are for 32^3 networks – Image decoder and inner-loop CNN parameter counts increase negligibly for higher resolutions.

6.4. Dataset

We conduct experiments on the 13 categories of the popular Shapenet dataset [6] popularized by Choy *et al*. [9]. Due to difficulties reconciling the rendering parameters, images and models supplied by the authors, we use our own renderings and voxelizations. As per Choy *et al*. each model was rendered from 24 different camera positions with azimuth angle uniformly sampled from $[0°, 360°)$ and elevation angle in $[25°, 30°]$ with resolution 128×128.

We created voxel grids by defining any voxel intersected by a face as filled. This means thin structures take up disproportionately large volumes at low resolutions. This is different to approaches which take a less strict approach which may preserve a better overall volume ratio but risk losing thin structures entirely. This difference can affect low resolution grids significantly, though the difference becomes insignificant at higher resolutions. After initial voxelization, grids were filled in consistent with the approach used by Johnston *et al*. [23].

6.5. Results

Images of two of our models' inferences are shown in Figure 6 – our smaller model trained with α-balanced cross-entropy loss and the larger model with continuous IoU. Unsurprisingly both models learn to space-carve very well, featuring virtually no voxels along rays that miss the object. The IoU-trained model appears to be more conserva-

	Frustum Dataset				World Aligned Dataset				
	IGE		Base		Base				
	MN	I4	MN	IF	MN	I4	R2N2 [9]	OGN [45]	Mat. [39]
plane	59.6	62.4	49.2	50.2	55.0	62.6	51.3	58.7	64.7
bench	52.4	55.2	47.3	47.9	52.8	58.1	42.1	48.1	57.7
cabinet	73.6	74.9	70.6	71.3	72.1	74.9	71.6	72.9	77.6
car	78.4	79.9	74.2	73.5	77.2	76.9	79.8	81.6	85.0
telephone	69.9	72.2	65.4	64.5	70.9	70.3	66.1	70.2	75.6
chair	57.0	60.1	52.4	53.6	55.0	60.7	46.6	48.3	54.7
sofa	69.6	71.2	65.9	66.8	66.7	69.8	62.8	64.6	68.1
rifle	60.6	62.6	47.8	50.0	55.0	60.2	54.4	59.3	61.6
lamp	54.0	56.5	47.5	50.1	48.7	50.8	38.1	39.8	40.8
monitor	58.8	60.7	53.5	55.4	54.7	60.0	46.8	50.2	53.2
speaker	74.5	76.5	72.4	72.8	70.6	72.4	66.2	63.7	70.1
table	57.4	60.6	52.9	54.3	57.8	61.0	51.3	53.6	57.3
watercraft	61.9	64.0	55.5	56.6	54.8	60.0	51.3	63.0	59.1
mean	63.7	**65.9**	58.0	59.0	60.8	**64.4**	56.0	59.5	63.5

Table 3: IoU values (in %) at 32^3 resolutions. IGE models was trained with continuous IoU loss from Equation 12. Mean values are calculated by class. A single model was trained across all categories for each of our columns.

tive when it comes to thin-structures, while the α-balanced model inferences often display slight shadowing along rays. This often results in more realistic looking inferences despite a lower average IoU score.

Quantitatively, we first investigate the performance of the models and effect of the frustum grid at 32^3 resolutions. We compare against R2N2 [9] – a standard benchmark – along with other approaches designed for high resolution inference: Octree-Generation Network (OGN) [45] and Matryoshka networks (Mat.) [39].

Intersection-over-union (IoU) values are shown in Table 3. Baseline models trained on the world-aligned grid consistently out-perform those trained on the frustum grid by a small margin. This suggests the patterns present in the frustum dataset are harder to learn than those in the regular dataset. This is not surprising, as there is significantly more variety in the frustum voxel grid dataset (1 grid per view, rather than 1 grid per model). For example, almost all planes in the world-aligned dataset have long fuselages and angled wings. A model that learns to identify planes could do reasonably well at low resolutions by simply inferring the class average rather than taking into account fine-level detail. To do similarly well on the frustum dataset, the model would need to additionally infer the camera position and learn to transform the average grid values accordingly.

While this means subsequent comparison to other methods trained on world-aligned grids is not truly fair, we include their results anyway. We believe this is more informative than only using self-comparisons so long as they are intepreted with this disclaimer in mind.

Our multi-level optimization approach clearly outperforms the baseline on the same dataset across all categories and both image networks. It also out-performs the base method on the easier world-aligned dataset, along with all other competing methods considered on average.

	IGE-MN				IGE-I4			
	cont. IoU	$\alpha=\gamma=0$	$\alpha=0.7$	$\gamma=2$	cont. IoU	$\alpha=\gamma=0$	$\alpha=0.7$	$\gamma=2$
32^3	**63.5**	60.7	58.0	59.8	**66.0**	61.9	59.6	64.0
64^3	**61.5**	56.8	57.1	56.9	**64.7**	60.8	59.3	59.3
128^3	**58.9**	51.6	54.5	53.9	**62.2**	56.4	56.8	57.1

Table 4: Mean IoU (in %, averaged over categories) for our IGE models trained with different losses.

	car				plane				table			
Resolution	32	64	128	256	32	64	128	256	32	64	128	256
OGN_1 [45]	64.1	77.1	78.2	76.6	-	-	-	-	-	-	-	-
MAT_1 [39]	68.3	78.4	79.4	**79.6**	36.7	48.8	58.0	59.6	38.6	42.3	43.5	41.3
$IGE\text{-}MN_{13}$	57.8	68.8	72.8	73.3	29.6	44.8	52.9	54.4	33.6	44.0	47.8	48.2
$IGE\text{-}I4_{13}$	57.9	70.9	74.0	75.2	30.5	47.8	57.5	57.3	34.8	46.5	52.7	50.5
$IGE\text{-}MN_1$	57.0	70.3	76.2	75.2	30.7	47.9	58.7	58.1	33.6	45.9	50.6	50.2
$IGE\text{-}I4_1$	58.4	71.2	76.5	76.5	30.1	49.2	60.5	**62.0**	35.0	46.4	52.2	**52.1**

Table 5: Mean IoU (in %) trained at difference resolutions and evaluated at 256^3 for models trained across all categories (13) and per-category (1). Per-category break-down of 13-category models available in supplementary.

Unsurprisingly, our larger model out-performs the smaller one in all categories, regardless of the model architecture.

To better understand the effect of the loss functions involved, we trained models at various resolutions with continous IoU loss compared to models trained with different versions of Equation 11: base cross entropy ($\alpha=0, \gamma=0$), reweighted cross entropy ($\alpha=0.7$, $\gamma=0$) and focal loss ($\alpha=0$, $\gamma=2$). Results are provided in Table 4.

Continuous IoU loss gives superior metrics scores to all variations of cross-entropy. There is no clear winner amongst cross-entropy variants.

Finally, we consider how our continuous IoU model performs at resolution of 256^3. Results for models trained at different resolutions and then linearly interpolated are given in Table 5. We trained a single model on all 13 categories, as well as a separate model for each of cars, planes and tables for fair comparison with other work.

Our networks all perform comparably on cars and planes, with our larger network performing slightly better, and category-specific training also improving things slightly. We significantly out-perform other methods on the table category, where the space-carving ability of our network can extract high-precision corners and edges and accurately reconstruct many thin structures.

Poor performance of low resolution models when evaluated at high resolutions is clear for our models. We attribute this to the large change in the volume of these structures as resolution increases as a result of our voxelization strategy.

A small performance regression is observed going from 128^3 to 256^3 in most experiments on our 13-category models. This is consistent with the observation made in OGN [45], who demonstrate that training on a more limited dataset results in improved performance with resolution, while more varied datasets are hindered by increased res-

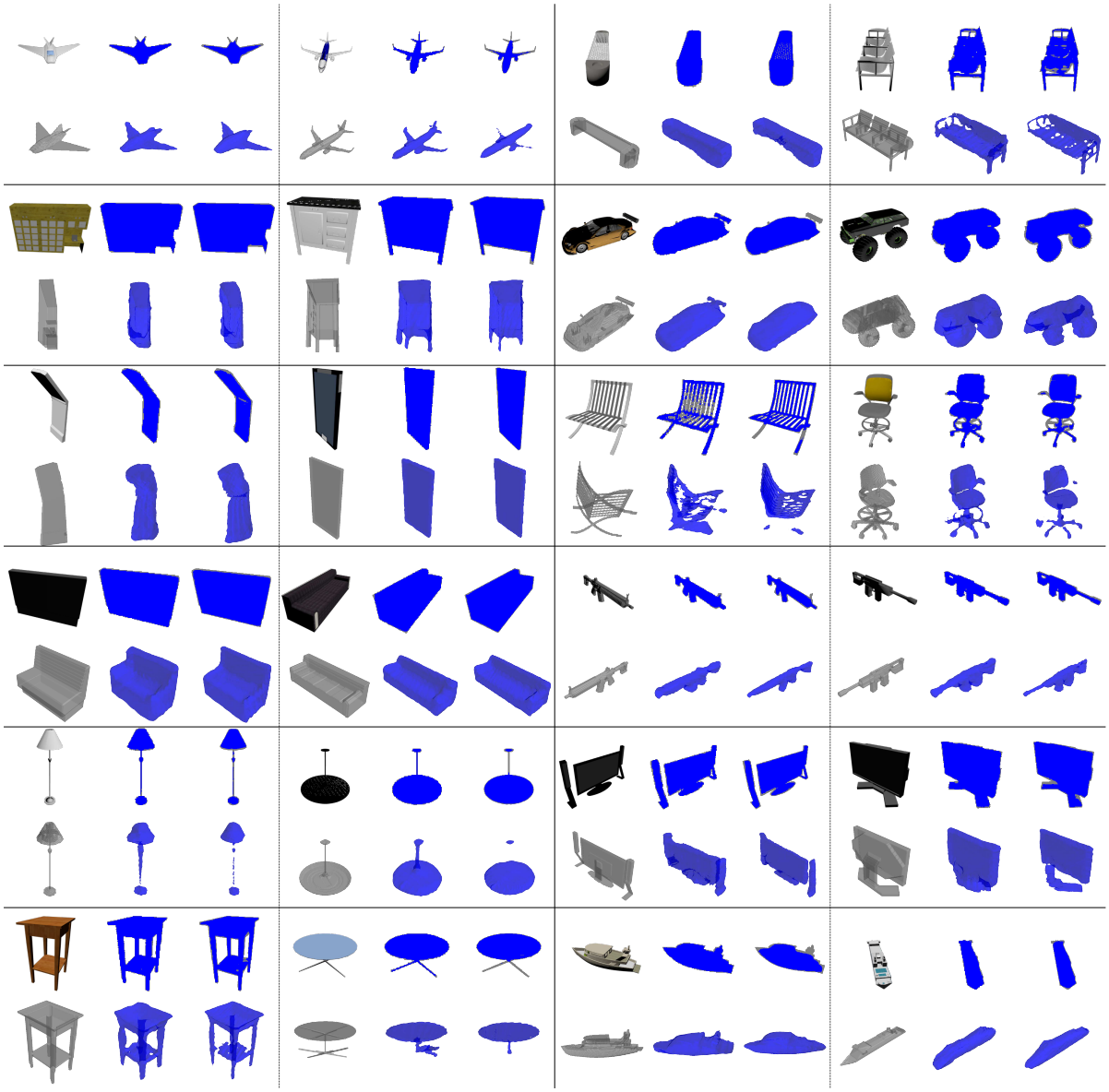

Figure 6: Sample results for IGE-MN model trained with $\alpha = 0.7$ loss @ 128^3 resolution and IGE-I4 model trained with continuous IoU @ 256^3. For each block of 6, top row: (left) input image, (middle) inference (blue) projected ground truth silhouettes (gray), (right) same as middle except I4 network trained with cont. IoU loss. Bottom row: (left) ground truth object, (middle) MN inference, (right) I4 inference.

olution. Unlike OGN, our regression occurs when training on the cross-category dataset, where as theirs is apparent training on the cars dataset.

7. Conclusion

We have demonstrated energy-based multi-level optimization networks can take advantage of computer graphics principles to infer 3D information from 2D inputs. Our human pose dimension-lifting model performed comparably to networks with orders of magnitude mode parameters and with a fraction of the number of operations. We investigated two 3D reconstruction networks, and showed competitive results could be achieved with a relatively small network, and a larger network could out-perform other state-of-the-art high-resolution networks.

This research was supported by the Australian Research Council through the grant ARC FT170100072.

References

[1] B. Amos and J. Z. Kolter. Optnet: Differentiable optimization as a layer in neural networks. *arXiv preprint arXiv:1703.00443*, 2017. 2

[2] D. Belanger and A. McCallum. Structured prediction energy networks. In *International Conference on Machine Learning*, pages 983–992, 2016. 2

[3] D. Belanger, B. Yang, and A. McCallum. End-to-end learning for structured prediction energy networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 429–439. JMLR.org, 2017. 2

[4] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016. 2

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 2

[6] A. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository., corr abs/1512.03012. *URL http://arxiv. org/abs/1512.03012*. 6

[7] C.-H. Chen and D. Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *CVPR*, volume 2, page 6, 2017. 2

[8] I. Cherabier, C. Häne, M. R. Oswald, and M. Pollefeys. Multi-label semantic 3D reconstruction using voxel blocks. In *3DV*, 2016. 2

[9] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 2, 6, 7

[10] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4715–4723, 2016. 2

[11] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2017. 2

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 11

[13] J. Domke. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*, pages 318–326, 2012. 2

[14] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *CVPR*, 2017. 2

[15] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016. 2

[16] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010. 4, 11

[17] J. Gwak, C. B. Choy, A. Garg, M. Chandraker, and S. Savarese. Weakly supervised generative adversarial networks for 3D reconstruction. In *3DV*, 2017. 2

[18] C. Häne, S. Tulsiani, and J. Malik. Hierarchical surface prediction for 3D object reconstruction. In *3DV*, 2017. 2

[19] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2014. 2, 4

[20] D. J. Rezende, S. M. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3D structure from images. In *NIPS*, 2016. 2

[21] D. Jack, F. Maire, A. Eriksson, and S. Shirazi. Adversarially parameterized optimization for 3d human pose estimation. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017. 2

[22] D. Jack, J. K. Pontes, S. Sridharan, C. Fookes, S. Shirazi, F. Maire, and A. Eriksson. Learning free-form deformations for 3d object reconstruction. *arXiv preprint arXiv:1803.10932*, 2018. 2

[23] A. Johnston, R. Garg, G. Carneiro, I. D. Reid, and A. van den Hengel. Scaling cnns for high resolution volumetric reconstruction from a single image. In *ICCV Workshops*, pages 930–939, 2017. 2, 6

[24] A. Kar, C. Häne, and J. Malik. Learning a multi-view stereo machine. In *NIPS*, 2017. 2

[25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 11

[26] A. Kurenkov, J. Ji, A. Garg, V. Mehta, J. Gwak, C. B. Choy, and S. Savarese. DeformNet: Free-form deformation network for 3d shape reconstruction from a single image. volume abs/1708.04672, 2017. 2

[27] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 1, 2

[28] C.-H. Lin, C. Kong, and S. Lucey. Learning efficient point cloud generation for dense 3D object reconstruction. In *AAAI*, 2018. 2

[29] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 6

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 4

[31] J. Liu, F. Yu, and T. A. Funkhouser. Interactive 3D modeling with a generative adversarial network. In *3DV*, 2017. 2

[32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 6

[33] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *International Conference on Computer Vision*, volume 1, page 5, 2017. 2, 3, 4, 5

[34] F. Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1561–1570. IEEE, 2017. 3

[35] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016. 2, 4, 6

[36] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *CVPR*, 2017. 2

[37] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view CNNs for object classification on 3D data. In *CVPR*, 2016. 2

[38] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 2

[39] S. R. Richter and S. Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1936–1944, 2018. 2, 6, 7

[40] G. Riegler, A. O. Ulusoy, and A. Geiger. OctNet: Learning deep 3D representations at high resolutions. In *CVPR*, 2017. 2

[41] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 6

[42] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 6, 11

[43] A. Sharma, O. Grau, and M. Fritz. VConv-DAE: Deep volumetric shape learning without object labels. In *ECCVW*, 2016. 2

[44] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. 6

[45] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, volume 2, page 8, 2017. 2, 7

[46] S. Tulsiani, A. A. Efros, and J. Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Computer Vision and Pattern Regognition (CVPR)*, 2018. 2

[47] H.-Y. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'17)*, volume 2, 2017. 2

[48] A. O. Ulusoy, A. Geiger, and M. J. Black. Towards probabilistic volumetric reconstruction using ray potential. In *3DV*, 2015. 2

[49] P.-S. Wang, Y. Liu, Y.-X. Guo, C.-Y. Sun, and X. Tong. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. In *SIGGRAPH*, 2017. 2

[50] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016. 2

[51] J. Wu, Y. Wang, T. Xue, X. Sun, W. T. Freeman, and J. B. Tenenbaum. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In *NIPS*, 2017. 2

[52] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Single image 3D interpreter network. In *ECCV*, 2016. 2

[53] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 2, 6

[54] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*, 2015. 2

[55] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *NIPS*, 2016. 2

[56] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 2018. 2

[57] Y. Yang, C. Feng, Y. Shen, and D. Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2018. 2

[58] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011. 2

[59] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015. 1

[60] R. Zhu, H. K. Galoogahi, C. Wang, and S. Lucey. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *NIPS*, 2017. 2