Escaping Plato's Cave: 3D Shape From Adversarial Rendering

Philipp Henzler
p.henzler@cs.ucl.ac.uk

Niloy J. Mitra
n.mitra@cs.ucl.ac.uk

Tobias Ritschel
t.ritschel@ucl.ac.uk

University College London

Abstract

We introduce PLATONICGAN *to discover the 3D structure of an object class from an unstructured collection of 2D images, i. e., where no relation between photos is known, except that they are showing instances of the same category. The key idea is to train a deep neural network to generate 3D shapes which, when rendered to images, are indistinguishable from ground truth images (for a discriminator) under various camera poses. Discriminating 2D images instead of 3D shapes allows tapping into unstructured 2D photo collections instead of relying on curated (e. g., aligned, annotated, etc.) 3D data sets.*

To establish constraints between 2D image observation and their 3D interpretation, we suggest a family of rendering layers *that are effectively differentiable. This family includes visual hull, absorption-only (akin to x-ray), and emission-absorption. We can successfully reconstruct 3D shapes from unstructured 2D images and extensively evaluate* PLATONICGAN *on a range of synthetic and real data sets achieving consistent improvements over baseline methods. We further show that* PLATONICGAN *can be combined with 3D supervision to improve on and in some cases even surpass the quality of 3D-supervised methods.*

1. Introduction

A key limitation to current generative models [37, 36, 12, 24, 32, 31] is the availability of suitable training data (e. g., 3D volumes, feature point annotations, template meshes, deformation prior, structured image sets, etc.) for supervision.

While methods exist to learn the 3D structure of classes of objects, they typically require 3D data as input. Regrettably, such 3D data is difficult to acquire, in particular for the "long tail" of exotic classes: ShapeNet might have `chair`, but it does not have `chanterelle`.

Addressing this problem, we suggest a method to learn 3D structure from 2D images only (Fig. 1). Reasoning about the 3D structure from 2D observations without assuming anything about their relation is challenging as illustrated by Plato's Allegory of the Cave [34]: *How can we hope to understand higher dimensions from only ever seeing projections?* If multiple views (maybe only two [40, 13]) of the same object are available, multi-view analysis without 3D supervision has been successful. Regrettably, most photo collections do not come in this form but are now and will remain *unstructured*: they show random instances under random pose, uncalibrated lighting in unknown relations, and multiple views of the same objects are not available.

Our first main contribution (Sec. 3) is to use adversarial training of a 3D generator with a discriminator that operates exclusively on widely available unstructured collections of 2D images, which we call *platonic discriminator*. Here, during training, the generator produces a 3D shape that is projected (rendered) to 2D and presented to the 2D Platonic discriminator. Making a connection between the 3D generator and the 2D discriminator, our second key contribution, is enabled by a family of *rendering layers* that can account for occlusion and color (Sec. 4). These layers do not need any learnable parameters and allow for backpropagation [26]. From these two key blocks we construct a system that learns

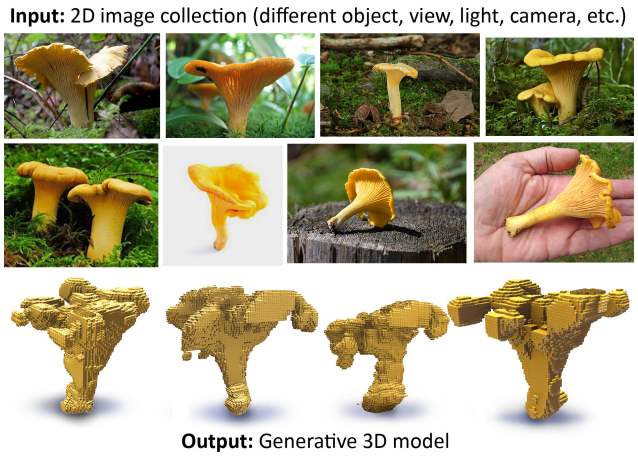


Figure 1. PLATONICGANs allow converting an unstructured collection of 2D images of a rare class (subset shown on top) into a generative 3D model (random samples below).

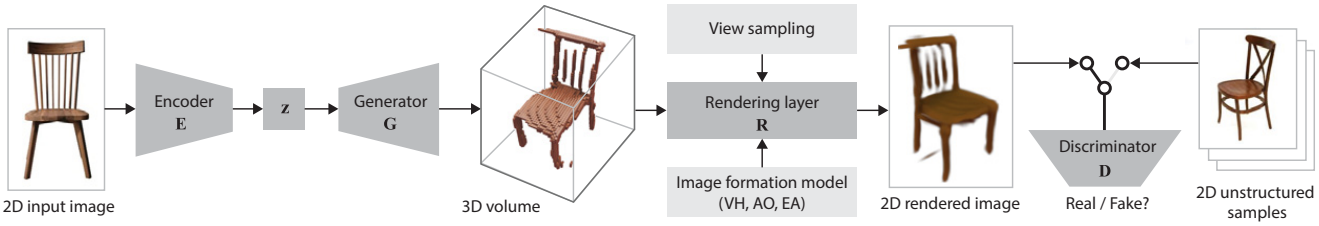


Figure 2. Overview: We encode a 2D input image using an encoder E into a latent code $\mathbf{z}$ and feed it to a generator G to produce a 3D volume. This 3D volume is inserted into a rendering layer R to produce a 2D rendered image which is presented to a discriminator D. The rendering layer is controlled by an image formation model: visual hull (VH), absorption-only (AO) or emission-absorption (EA) and view sampling. The discriminator D is trained to distinguish such rendered imagery from an unstructured 2D photo collection, i. e., images of the same class of objects, but not necessarily having repeated instances, view or lighting and with no assumptions about their relation (e.g., annotated feature points, view specifications).

the 3D shapes of common classes such as chairs and cars, but also exotic classes from unstructured 2D photo collections. We demonstrate 3D reconstruction from a single 2D image as a key application (Sec. 5). While recent works focus on using as little explicit supervision [17, 19, 8, 29, 28, 11] as possible, they all rely on either annotations, 3D templates, known camera poses, specific views or multi-view images during training. Our approach takes it a step further by receiving no such supervision, see Tbl. 1.

Table 1. Taxonomy of different methods that learn 3D shapes with no explicit 3D supervision. We compare Kanazawa et al. [17], Kato et al. [19], Eslami et al. [8], Tulsiani et al. [29], Tulsiani et al. [28], PrGan [11] with our method in terms of degree of supervision.

Supervision at training time	[17]	[19]	[8]	[29]	[28]	[11]	Ours
Annotation-free	×	✓	✓	✓	✓	✓	✓
3D template-Free	×	×	✓	✓	✓	✓	✓
Unknown camera pose	✓	×	×	×	✓	✓	✓
No pre-defined camera poses	✓	✓	✓	✓	×	×	✓
Only single view required	✓	×	×	×	×	✓	✓
Color	✓	✓	✓	✓	×	×	✓

2. Related Work

Several papers suggest (adversarial) learning using 3D voxel representations [37, 36, 12, 24, 11, 32, 31, 35, 39, 30, 20] or point cloud input [1, 10]. The general design of such networks is based on an encoder that generates a latent code which is then fed into a generator to produce a 3D representation (i. e., a voxel grid). A 3D discriminator now analyzes samples both from the generator and from the ground truth distribution. Note that this procedure requires 3D supervision, i. e., is limited by the type and size of the 3D data set such as ShapeNet [5].

Girdhar et al. [12] work on a joint embedding of 3D voxels and 2D images, but still require 3D voxelizations as input. Fan et al. [9] produce points from 2D images, but similarly with 3D data as training input. Gadelhan et al. [11] use 2D visual hull images to train a generative 3D model. Cho et al.'s recursive design takes multiple images as input [6] while also being trained on 3D data. Kar et al. [18] propose a simple "unprojection" network component to establish a relation between 2D pixels and 3D voxels but without resolving occlusion and again with 3D supervision.

Cashman and Fitzgibbon [4] and later Carreira et al. [3] or Kanazawa et al. [17] use correspondence to 3D templates across segmentation- or correspondence-labeled 2D image data sets to reconstruct 3D shapes. These present stunning results, for example on animals, but at the opposite end of a spectrum of manual human supervision, in which our approach receives no such supervision.

Closer to our approach is Rezende et al. [25] that also learn 3D representations from single images. However, they make use of a partially differentiable renderer [22] that is limited to surface orientation and shading, while our formulation can resolve both occlusion from the camera and appearance. Also, their representation of the 3D volume is a latent one, that is, it has no immediate physical interpretation that is required in practice, e. g., for measurements, to run simulations such as renderings or 3D printing. This choice of having a deep representation of the 3D world is shared by Eslami et al. [8]. Tulsiani et al. [29] reconstruct 3D shape supervised by multiple 2D images of the same object with known view transformations at learning time. Tulsiani et al. [28] take it a step further and require no knowledge about the camera pose, but still require multiple images of the same object at training time. They have investigated modelling image formation as sums of voxel occupancies to predict termination depth. We use a GAN to train on photo collections which typically only show one view of each instance. Closest to our work is Gadelha et al. [11] which operates on an unstructured set of visual hull images but receives three sources of supervision: view information gets explicitly encoded as a dimension in the latent vector; views come from a manually-chosen 1D subspace (circle); and there are only 8 discrete views. We take the image formation a step further to support absorption-only and emission-absorption

image formation, allowing to learn from real photos and do so on unstructured collections from-the-wild where no view supervision is available.

While early suggestions how to extend differentiable renderers to polygonal meshes exist, they are limited to deformation of a pre-defined template [19]. We work with voxels, which can express arbitrary topology, e. g., we can generate chairs with drastically different layout, which are not a mere deformation of a base shape.

Similarly, inter-view constraints can be used to learn depth maps [40, 13] using reprojection constraints: If the depth label is correct, reprojecting one image into the other view has to produce the other image. Our method does not learn a single depth map but a full voxel grid and allows principled handling of occlusions.

A generalization from visual hull maps to full 3D scenes is discussed by Yan et al. [38]. Instead of a 3D loss, they employ a simple projection along major axis allowing to use a 2D loss. However, multiple 2D images of the same object are required. In practice this is achieved by rendering the 3D shape into 2D images from multiple views. This makes two assumptions: We have multiple images in a *known* relation and available reference appearance (i. e., light, materials). Our approach relaxes those two requirements: we use a discriminator that can work on arbitrary projections and arbitrary natural input images, without known reference.

3. 3D Shape From 2D Photo Collections

We now introduce PLATONICGAN (Fig. 2). The rendering layers used here will be introduced in Sec. 4.

Common GAN Our method is a classic (generative) adversarial design [14] with two main differences: The discriminator D operates in 2D while the 3D generator G produces 3D output. The two are linked by a fixed-function projection operator, i. e., non-learnable (see Sec. 4).

Let us recall the classic adversarial learning of 3D shapes [36], which is a min-max game

$$\min_{\Theta} \max_{\Psi} \; c_{\text{Dis}}(\Psi) + c_{\text{Gen}'}(\Theta) \tag{1}$$

between the discriminator and the generator cost, respectively c_{Dis} and $c_{\text{Gen}'}$.

The discriminator cost is

$$c_{\text{Dis}}(\Psi) = \mathbb{E}_{p_{\text{Data}}(\mathbf{x})}[\log(D_\Psi(\mathbf{x}))] \tag{2}$$

where D_Ψ is the discriminator with learned parameters Ψ which is presented with samples $\mathbf{x}$ from the distribution of real 3D shapes $\mathbf{x} \sim p_{\text{Data}}$. Here $\mathbb{E}_p$ denotes the expected value of the distribution p.

The generator cost is

$$c_{\text{Gen}'}(\Theta) = \mathbb{E}_{p_{\text{Gen}}(\mathbf{z})}[\log(1 - D_\Psi(G_\Theta(\mathbf{z}))] \tag{3}$$

where G_Θ is the generator with parameters Θ that maps the latent code $\mathbf{z} \sim p_{\text{Gen}}$ to the data domain.

PLATONICGAN The discriminator cost is calculated identical to the common GAN with the only difference that the input samples are rendered 2D images with generation cost

$$c_{\text{Gen}}(\Theta) = \mathbb{E}_{p_{\text{Gen}}(\mathbf{z})}\mathbb{E}_{p_{\text{View}}(\omega)}[\log(1 - D_\Psi(R(\omega, G_\Theta(\mathbf{z}))))], \tag{4}$$

where R projects the generator result $G_\Theta(\mathbf{z})$ from 3D to 2D along the sampled view direction ω. See Sec. 3.1 for details.

While many parameterizations for views are possible, we choose an orthographic camera with fixed upright orientation that points at the origin from an Euclidean position $\omega \in \mathbb{S}^2$ on the unit sphere. $\mathbb{E}_{p_{\text{View}}(\omega)}$ is the expected value across the distributions $\omega \sim p_{\text{View}}$ of views.

PLATONICGAN 3D Reconstruction Two components in addition to our Platonic concept are required to allow for 3D reconstruction, resulting in

$$\min_{\Psi} \max_{\Theta,\Phi} \; c_{\text{Disc}}(\Psi) + c_{\text{Gen}}(\Theta, \Phi) + \lambda c_{\text{Rec}}(\Theta, \Phi), \tag{5}$$

where c_{Gen} includes an encoding step and c_{Rec} encourages the encoded generated-and-projected result to be similar to the encoder input where $\lambda = 100$. We detail both of these steps in the following paragraphs:

Generator The generator G_Θ does not directly work on a latent code $\mathbf{z}$, but allows for an encoder E_Φ with parameters Φ that encodes a 2D input image $\mathbf{I}$ to a latent code $\mathbf{z} = E_\Phi(\mathbf{I})$. The cost becomes,

$$c_{\text{Gen}}(\Theta, \Phi) = \mathbb{E}_{p_{\text{Dat}}(\mathbf{I})}\mathbb{E}_{p_{\text{View}}(\omega)}[\log(1 - D_\Psi(R(\omega, G_\Theta(E_\Phi(\mathbf{I})))))]. \tag{6}$$

Reconstruction We encourage the encoder E_Φ and generator G_Θ to reproduce the input in the $\mathcal{L}_2$ sense: by convention the input view is $\omega_0 = (0, 0)$,

$$c_{\text{Rec}}(\Theta, \Phi) = \|\mathbf{y} - R(\omega_0, G_\Theta(E_\Phi(\mathbf{I})))\|_2^2 \tag{7}$$

where $\mathbf{y}$ represents the ground truth image. While this step is not required for generation it is mandatory for reconstruction. Furthermore, it adds stability to the optimization as it is easy to find an initial solution that matches this 2D cost before refining the 3D structure.

3.1. Optimization

Two key properties are essential to successfully optimize our PLATONICGAN: First, maximizing the expected value across the distribution of views p_{View} and second, back-propagation through the projection operator R. We extend the classic GAN optimization procedure in Alg. 1.

Algorithm 1 PLATONICGAN Reconstruction Update Step

$I_{\text{Dat}} \leftarrow \text{SAMPLEIMAGE}(p_{\text{Dat}})$
$\omega \leftarrow \text{SAMPLEVIEW}(p_{\text{View}})$
$z \leftarrow \text{E}(I_{\text{Dat}})$
$v \leftarrow \text{G}(z)$
$I_{\text{View}} \leftarrow \text{R}(\omega, v)$
$I_{\text{Front}} \leftarrow \text{R}(\omega_0, v)$
$c_{\text{Dis}} \leftarrow \log D(I_{\text{Dat}}) + \log(1 - D(I_{\text{View}}))$
$c_{\text{Gen}} \leftarrow \log(1 - D(I_{\text{View}}))$
$c_{\text{Rec}} \leftarrow \text{L2}(I_{\text{Dat}} - I_{\text{Front}})$
$\Psi \leftarrow \text{MAXMIZE}(c_{\text{Dis}})$
$\Theta, \Phi \leftarrow \text{MINIMIZE}(c_{\text{Gen}} + \lambda c_{\text{Rec}})$

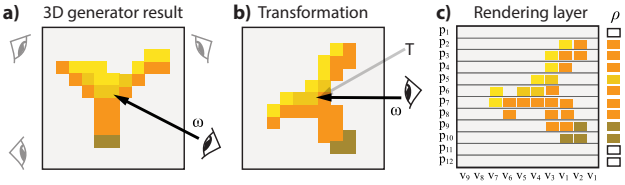


Figure 3. Rendering layers (Please see text).

Figure 4. Different image formation models visual hull (VH), absorption-only (AO) and emission-absorption (EA).

Projection We focus on the case of a 3D generator on a regular voxel grid $\mathbf{v}^{n_\text{c} \times n_\text{p}^3}$ and a 2D discriminator on a regular image $\mathbf{I}^{n_\text{c} \times n_\text{p}^2}$ where n_c denotes the number of channels and $n_\text{p} = 64$ corresponds to the resolution. In section 4, we discuss three different projection operators. We use $R(\omega, \mathbf{v})$ to map a 3D voxel grid $\mathbf{v}$ under a view direction $\omega \in \mathbb{S}^2$ to a 2D image $\mathbf{I}$.

We further define $R(\omega, \mathbf{v}) := \rho(\mathsf{T}(\omega)\mathbf{v})$ with rotation matrix $\mathsf{T}(\omega)$ according to the view direction ω and an image formation function $\rho(\mathbf{v})$ that is view-independent. The same transformation is shared by all implementations of the rendering layer, so we will only discuss the key differences of ρ in the following. Note that a rotation and a linear resampling is back-propagatable and typically provided in a deep learning framework, e. g., as `torch.nn.functional.grid_sample` in PyTorch [23]. While we work in orthographic space, ρ could also model a perspective transformation.

View sampling We assume uniform view sampling.

4. Rendering Layers

Rendering layers (Fig. 3) map 3D information to 2D images so they can be presented to a discriminator. We first assume the 3D volume to be rotated (Fig. 3, a) into camera space from view direction ω (Fig. 3, b), such that the pixel value p is to be computed from all voxel values $\mathbf{v}_i$ and only those (Fig. 3, c). The rendering layer maps a sequence of n_z voxels to a pixel value $\rho(\mathbf{v}) \in \mathbb{R}^{n_\text{c} \times n_\text{p}^3} \to \mathbb{R}^{n_\text{c} \times n_\text{p}^2}$. Composing the full image $\mathbf{I}$ just amounts to executing ρ for every pixel p resp. all voxels $\mathbf{v} = v_1, \ldots, v_{n_\text{z}}$ at that pixel.

Note, that the rendering layer does not have any learnable parameters. We will now discuss several variants of ρ, implementing different forms of volume rendering [7]. Fig. 4 shows the image formation models we currently support.

Visual hull (VH) Visual hull [21] is the simplest variant (Fig. 4). It converts scalar density voxels into binary opacity images. A voxel value of 0 means empty space and a value of 1 means fully occupied, i. e., $v_i \in [0, 1]$. Output is a binary value indicating if any voxel blocked the ray. It is approximated as

$$\rho_{\text{VH}}(\mathbf{v}) = 1 - \exp(\sum_i -v_i). \tag{8}$$

Note that the sum operator can both be back-propagated and is efficiently computable on a GPU using a parallel scan. We can apply this to learn 3D structure from binary 2D data such as segmented 2D images.

Absorption-only (AO) The absorption-only model is the gradual variant of visual hull. This allows for "softer" attenuation of rays. It is designed as:

$$\rho_{\text{AO}}(\mathbf{v}) = 1 - \prod_i (1 - v_i). \tag{9}$$

If v_i are fractional the result is similar to an x-ray, i. e., $v_i \in [0, 1]$. This image formation allows learning from x-rays or other transparent 2D images. Typically, these are single-channel images, but a colored variant (e. g., x-ray at different wavelength or RGB images of colored transparent objects) could technically be done.

Emission-absorption (EA) Emission-absorption allows the voxels not only to absorb light coming towards the observer but also to emit new light at any position. This interplay of emission and absorption can model occlusion, which we will see is useful to make 3D sense of a 3D world. Fig. 3 uses emission-absorption with high absorption, effectively realizing an opaque surface with visibility.

A typical choice is to have the absorption v_a monochromatic and the emission v_e chromatic.

The complete emission-absorption equation is

$$\rho_{\text{EA}}(\mathbf{v}) = \sum_{i=1}^{n_\text{z}} \underbrace{(1 - \prod_{j=1}^{i} (1 - v_{\text{a},j}))}_{\text{Transmission } t_i \text{ to voxel } i} v_{\text{e},i} \tag{10}$$

While such equations are typically solved using ray-marching [7], they can be rewritten to become differentiable in practice: First, we note that the transmission t_i from voxel i is one minus a product of one minus the density of all voxels before i. Similar to a sum such a cumulative product can be back-propagated and computed efficiently using parallel scans, e. g., using `torch.cumprod`. A numerical alternative, that performed similar in our experiments, is to work in the log domain and use `torch.cumsum`.

5. Evaluation

Our evaluation comprises of a quantitative (Sec. 5.4) and a qualitative analysis (Sec. 5.5) that compares different previous techniques and ablations to our work (Sec. 5.2).

5.1. Data sets

Synthetic We evaluate on two synthetic data sets: (a) ShapeNet [5] and (b) mammalian skulls [16]. For our quantitative analysis, we use ShapeNet models as 3D ground truth is required, but strictly only for evaluation, never in our training. 2D images of 3D shapes are rendered for the three image formation models VH, AO, EA. Each shape is rendered from a random view (50 per object), with random natural illumination. ShapeNet only provides 3D density volumes which is not sufficient for EA analysis. To this end, we use volumetric projective texturing to propagate the appearance information from thin 3D surface crust as defined by ShapeNet's textures into the 3D voxelization in order to retrieve RGBA volumes where A corresponds to density. We use shapes from the classes `airplane`, `car`, `chair`, `rifle` and `lamp`. The same train / validation / test split as proposed by [5] is adopted.

We also train on a synthetic x-ray data set that consists of 466,200 mammalian skull x-rays [16]. We used the monkey skulls subset of that data set (∼30k x-rays).

Real We use two data sets of rare classes: (a) `chanterelle` (60 images) and (b) `tree` (37 images) (not strictly rare, but difficult to 3D-model). These images are RGBA, masked, on white background. Note, that results on these input data has to remain qualitative, as we lack the 3D information to compare to and do not even have a second view of the same object to even perform an image comparison.

5.2. Baselines and comparison

2D supervision First, we compare the publicly available implementation of PrGAN [11] with our Platonic method. PrGAN is trained on an explicitly created data set adhering to their view restrictions (8 views along a single axis). Compared to our method it is only trained on visual hull images, however for evaluation purposes absorption-only and emission-absorption (in form of luminance) images are used as input images at test time. Note that PrGAN allows for object-space view reconstruction due to view information in the latent space whereas our method performs reconstruction in view-space. Due to the possible ambiguities in the input images (multiple images can belong to the same 3D volume), the optimal transformation into object space is found using a grid search across all rotations.

3D supervision The first baseline with 3D supervision is MULTI-VIEW, that has training-time access to multiple images of the same object [38] in a known spatial relation. Note, that this is a stronger requirement than for PLATONICGAN that does not require any structure in the adversarial examples: geometry, view, light – all change, while in this method only the view changes in a prescribed way.

The second competitor is a classic 3DGAN [36] trained with a Wasserstein loss [2] and gradient penalty [15].

To compare PLATONICGAN against methods having access to 3D information, we also propose a variant PLATONIC3D by adding the PLATONICGAN adversarial loss term (for all images and shapes) to the 3DGAN framework.

5.3. Evaluation Metrics

2D evaluation measures Since lifting 2D information to 3D can be ambiguous, absolute 3D measures might not be the best suitable measures for evaluation on our task. For instance, a shift in depth of an object under an orthographic camera assumption will result in a higher error for metrics in 3D, but the shift would not have any effect on a rendered image. Thus, we render both the reconstructed and the reference volume from the same 10 random views and compare their images using SSIM / DSSIM [33] and VGG16 [27] features. For this re-rendering, we further employ four different rendering methods: the original (i. e., ρ) image formation (IF), volume rendering (VOL), iso-surface rendering with an iso-value of .1 (ISO) and a voxel rendering (VOX), all under random natural illumination.

3D evaluation measures We report root-mean-squared-error (RMSE), intersection-over-union (IoU) and chamfer distance (CD). For the chamfer distance we compute a weighted directional distance:

$$d_{\mathrm{CD}}(T, O) = \frac{1}{N} \sum_{p_i \in T} \min_{p_j \in O} w_j \|\mathbf{p}_i - \mathbf{p}_j\|_2^2,$$

where T and O correspond to output and target volumes respectively, and w_j denotes the density value of the voxel at location $\mathbf{p}_j$. The weighting makes intuitive sense as our results have scalar values rather than binary values, i. e., higher densities get penalized more, and N is the total number of voxels in the volume. We give preference to such a weighting opposed to finding a threshold value for binarization.

Table 2. Performance of different methods with varying degrees of supervision (superv.) **(rows)** on different metrics **(columns)** for the class `airplane`. Evaluation is performed on all three image formations (IF): visual hull (VH), absorption-only (AO) and emission-absorption (EA). Note, DSSIM and VGG values are multiplied by 10, RMSE by 10^2 and CD by 10^3. Lower is better except for IoU.

Method	IF	Superv.		2D Image Re-synthesis										3D Volume			FID
				VH		AO		EA		VOX		ISO					EA
		2D	3D	DSSIM	VGG	DSSIM	VGG	DSSIM	VGG	DSSIM	VGG	DSSIM	VGG	RMSE	IoU	CD	
PrGAN [11]	VH	✓	×	1.55	6.57	1.37	**4.85**	1.41	**4.63**	1.68	5.41	1.83	6.15	**7.46**	0.11	**3.59**	207
Ours	VH	✓	×	**1.14**	**5.37**	**1.16**	4.93	**1.12**	4.68	**1.33**	**5.22**	**1.28**	**5.96**	9.16	**0.20**	11.77	**55**
Mult.-View [38]	VH	✓	×	0.87	4.89	0.80	4.31	0.90	4.07	1.38	4.83	1.21	5.56	5.37	0.36	**9.31**	155
3DGAN [36]	VH	✓	×	0.83	5.01	**0.75**	4.02	0.86	**3.83**	1.30	4.73	1.17	5.82	**4.97**	**0.46**	14.60	111
Ours 3D	VH	✓	×	**0.81**	**4.82**	0.77	**3.98**	**0.83**	**3.83**	**1.18**	**4.59**	**1.09**	**5.50**	5.20	0.44	12.33	**98**
PrGAN [11]	AO	✓	×	1.41	6.40	1.27	4.80	1.27	4.52	1.53	5.32	1.63	6.00	7.11	0.09	**2.78**	190
Ours	AO	✓	×	**0.94**	**5.35**	**0.93**	**4.46**	**0.91**	**4.26**	**1.11**	**4.96**	**1.09**	**5.75**	**5.70**	**0.27**	6.98	**90**
Mult.-View [38]	AO	✓	×	0.95	4.99	0.78	4.23	0.91	4.01	1.51	4.92	1.29	5.39	**4.89**	0.34	**9.47**	165
3DGAN [36]	AO	✓	×	0.67	4.37	0.69	3.77	0.72	3.57	0.99	**4.25**	0.97	**4.92**	5.08	**0.43**	14.92	**58**
Ours 3D	AO	✓	×	**0.66**	**4.36**	**0.66**	**3.73**	**0.70**	**3.52**	**0.98**	4.28	**0.96**	4.94	5.17	0.37	15.43	64
PrGAN [11]	EA	✓	×	**1.31**	**6.22**	**1.15**	**4.77**	**1.16**	**5.37**	**1.36**	**6.71**	**1.47**	**7.07**	**6.80**	0.08	**2.36**	196
Ours	EA	✓	×	2.18	6.53	1.99	5.38	1.89	6.00	2.21	7.43	2.36	7.92	14.13	**0.13**	10.53	**181**
Mult.-View [38]	EA	✓	×	1.62	6.21	1.53	4.58	1.63	5.48	1.95	6.97	1.94	7.41	15.05	0.12	32.07	172
3DGAN [36]	EA	✓	×	0.89	5.28	**0.78**	**3.93**	0.98	4.79	1.29	6.76	1.30	7.09	**5.24**	**0.46**	**13.66**	110
Ours 3D	EA	✓	×	**0.82**	**4.71**	0.82	3.96	**0.97**	**4.77**	**1.12**	**6.12**	**1.16**	**6.47**	7.43	0.04	18.82	**73**

5.4. Quantitative evaluation

Tbl. 2 summarizes our main results for the `airplane` class. Concerning the image formation models, we see that the overall values are best for AO, which is expected: VH asks for scalar density but has only a binary image; AO provides internal structures but only needs to produce scalar density; EA is hardest, as it needs to resolve both density and color. Nonetheless the differences between us and competitors are similar across the image formation models.

2D supervision We see that overall, our 2D supervised method outperforms PrGAN for VH and AO. Even though PrGAN was not trained on EA it wins for all metrics against our 2D supervised method. However, it even outperforms the 3D supervised methods 3DGAN and MULTI-VIEW which demonstrates the complexity of the task itself. However, PrGAN for EA only produces density volumes unlike all other methods that produce RGBA volumes. Comparing our 2D supervised method against the 3D supervised methods we see that overall our method produces competitive results. Regarding MULTI-VIEW we sometimes even perform better.

3D supervision Comparing our PLATONIC3D variant to the 3D baselines we observe our method to mostly outperform them for 2D metrics. Not surprisingly our method performs worse for 3D metrics as our approach only operates in 2D.

In Tbl. 3 we look into the performance across different classes. `rifle` performs best: the approach learns quickly from 2D that a gun has an outer 3D shape that is a revolute structure. `chair` performs worst, likely due to its high intra-class variation.

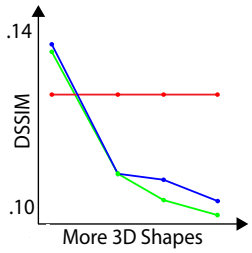
Table 3. Reconstruction performance of our method for different image formation models **(columns)** on different classes **(rows)**. The error metric is SSIM (higher is better).

Class	VH			AO			EA		
	VOL	ISO	VOX	VOL	ISO	VOX	VOL	ISO	VOX
`plane`	0.93	0.92	0.93	0.94	0.93	0.93	0.85	0.76	0.77
`rifle`	0.95	0.94	0.95	0.95	0.94	0.95	0.90	0.78	0.80
`chair`	0.86	0.85	0.85	0.86	0.85	0.86	0.80	0.61	0.63
`car`	.841	.846	.851	.844	.846	.850	.800	.731	.743
`lamp`	.920	.915	.920	.926	.914	.920	.883	.790	.803

In Tbl. 4 we compare the mean VGG error of a vanilla 3D GAN trained only on 3D shapes, a Platonic approach accessing only 2D images, and PLATONIC3D that has access to both. We keep the number of 2D images fixed, and increase the number of 3D shapes available; the horizontal axis in Tbl. 4. Without making use of the 3D supervision, the error of PLATONICGAN remains constant, independent of the number of 3D models. Like this, we see that a PLATONICGAN (red line) can beat both other approaches in a condition where little 3D data is available (left). When

Figure 5. Visual results for 3D reconstruction of three classes (`airplane`, `chair`, `rifle`) from multiple views.

Table 4. Effect of number of 3D shapes and 2D images on learning different methods in terms of mean DSSIM error. Lower is better.



2D images	70k	70k	70k	70k
3D shapes	5	50	250	1.5k
2D-3D ratio	14k	1.4k	280	47
● 3D	.135	**.108**	.106	.101
● Ours	**.125**	.125	.125	.125
● Ours 3D	.134	**.108**	**.102**	**.099**

more 3D data is available, PLATONICGAN (green line) wins over a pure 3D GAN (blue line). We conclude that adding 2D image information to a 3D corpus helps, and when the corpus is small enough even performs better than 3D-only supervised methods.

5.5. Qualitative

Synthetic Fig. 5 shows typical results for the reconstruction task. We see that our reconstruction can produce `airplane`, `chair` and `rifle` 3D models representative of the input 2D image. Most importantly, these 3D models look plausible for multiple views, not only from the input one. The results on the chair category also show that the model captures the relevant variation, ranging from straight chairs over club chairs to armchairs. For `gun`, the results turn out almost perfect, in agreement with the numbers reported before. In summary, our quality is comparable to GANs with 3D supervision.

2D vs. 3D vs. 2D+3D Qualitative comparison of 2D-only, 3D-only and mixed 2D-3D training can be seen in Fig. 6.

Synthetic rare We explored reconstructing skulls from x-ray (i. e., the AO IF model) images [16] in Fig. 9. We find the method to recover both external and internal structures.

Real rare Results for rare classes are seen in Fig. 1 and Fig. Fig. 7. We see that our method produces plausible details from multiple views while respecting the input image, even in this difficult case. No metric can be applied to these data as no 3D volume is available to compare in 3D or re-project.

6. Discussion

Why not having a multi-view discriminator? It is tempting to suggest a discriminator that does not only look at a single image, but at multiple views at the same time to judge if the generator result is plausible holistically. But while we can generate "fake" images from multiple views p_{Data}, the set of "real" natural images does not come in such a form. As a key advantage, our method only expects unstructured data: online repositories hold images with unknown camera, 3D geometry or illumination.

Failure cases are depicted in Fig. 8. Our method struggles to reconstruct the correct pose as lifting 2D images to 3D shapes is ambiguous for view-space reconstruction.

Supplemental More analysis, videos, training data and network definitions are available at `https://geometry.cs.ucl.ac.uk/projects/2019/platonicgan/`.

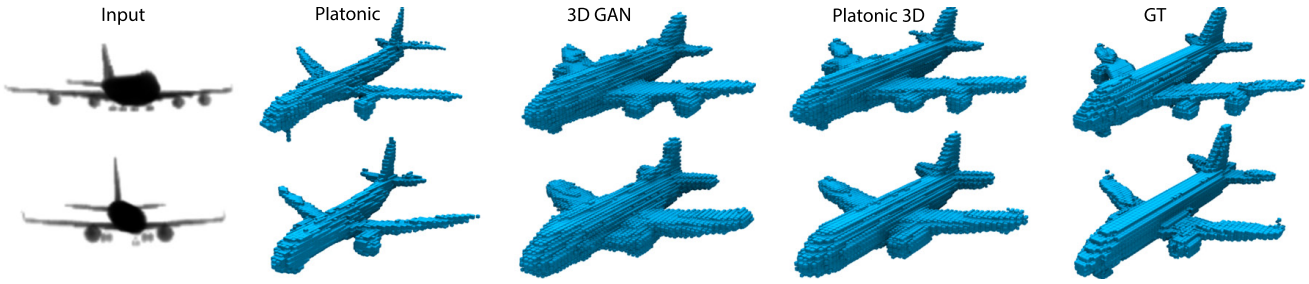


Figure 6. Comparison of 3D reconstruction results using the class `plane` between different forms of supervision **(columns)** for two different input views **(rows)**. PLATONICGAN, in the second column, can reconstruct a plausible plane, but with errors such as a wrong number of engines. The 3D GAN in the third column fixes this error, but at the expense of slight mode collapse where instances look similar and slightly "fat". Combining a 3D GAN with adversarial rendering as in the fourth row, is closest to the reference in the fifth row.

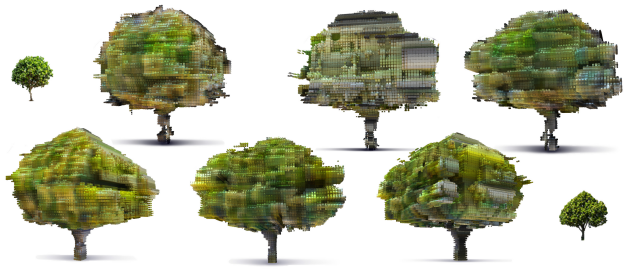


Figure 7. 3D Reconstruction of different trees using the emission-absorption image formation model, seen from different views **(columns)**. The small images were used as input. We see that PLATONICGAN has understood the 3D structure, including a distinctly colored stem, fractal geometry and structured leave textures.

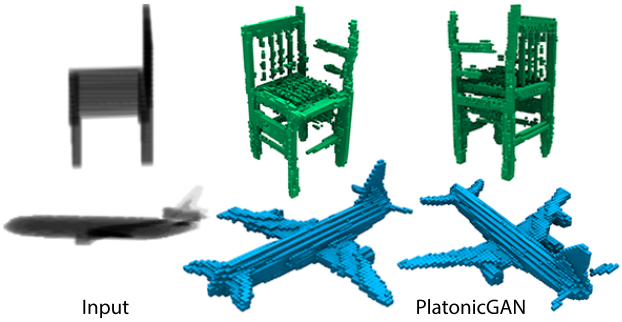


Figure 8. Failure cases of a chair **(top)** and an airplane **(bottom)**. The encoder is unable to estimate the correct camera-pose due to view-ambiguities in the input image and symmetries in the shapes. The generator then tries to satisfy multiple different camera-poses.

7. Conclusion

In this paper, we have presented PLATONICGAN, a new approach to learning 3D shapes from unstructured collections of 2D images. The key to our "escape plan" is to train a 3D generator outside the cave that will fool a discriminator seeing projections inside the cave.

We have shown a family of rendering operators that can be GPU-efficiently back-propagated and account for occlusion and color. These support a range of input modalities, ranging from binary masks, over opacity maps to RGB images with transparency. Our 3D reconstruction application is build on top of this idea to capture varied and detailed 3D shapes, including color, from 2D images. Training is exclusively performed on 2D images, enabling 2D photo collections to contribute to generating 3D shapes.

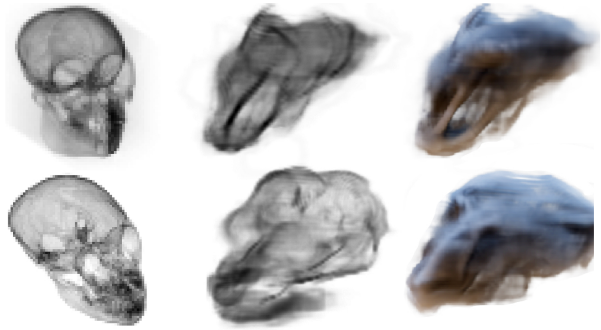


Figure 9. PlatonicGANs trained on 2D x-rays (i. e., AO IF) of mammalian skulls **(a)**. The resulting 3D volumes can be rendered from novel views using x-ray **(b)** and under novel views in different appearance, here, using image-based lighting **(c)**.

Future work could include shading that is related to gradients of density [7] into classic volume rendering. Furthermore, any sort of differentiable rendering operator ρ can be added. Devising such operators is a key future challenge. Other adversarial applications such as 2D supervised completion of 3D shapes seems worth exploring. Enabling object-space as opposed to view-space reconstruction would help to prevent failure cases as shown in Fig. 8.

While we combine 2D observations with 3D interpretations, similar relations might exist in higher dimensions, between 3D observations and 4D (3D shapes in motion) but also in lower dimensions, such as for 1D row scanner in robotics or 2D slices of 3D data such as in tomography.

Acknowledgements This work was supported by the ERC Starting Grant SmartGeometry, a GPU donation by NVIDIA Corporation and a Google AR/VR Research Award.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. 2018. 2

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 5

[3] Joao Carreira, Sara Vicente, Lourdes Agapito, and Jorge Batista. Lifting object detection datasets into 3d. *IEEE PAMI*, 38(7):1342–55, 2016. 2

[4] Thomas J Cashman and Andrew W Fitzgibbon. What shape are dolphins? building 3D morphable models from 2D images. *PAMI*, 35(1):232–44, 2013. 2

[5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv:1512.03012*, 2015. 2, 5

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *ECCV*, pages 628–44, 2016. 2

[7] Robert A Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. In *Siggraph Computer Graphics*, volume 22, pages 65–74, 1988. 4, 5, 8

[8] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–10, 2018. 2

[9] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3D object reconstruction from a single image. *arXiv:1612.00603*, 2016. 2

[10] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, volume 2, page 6, 2017. 2

[11] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *3DV*, 2016. 2, 5, 6

[12] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, pages 484–99, 2016. 1, 2

[13] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, pages 6602–6611, 2017. 1, 3

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–80, 2014. 3

[15] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NIPS*, pages 5767–5777, 2017. 5

[16] Philipp Henzler, Volker Rasche, Timo Ropinski, and Tobias Ritschel. Single-Image Tomography: 3D Volumes from 2D Cranial X-Rays. *Computer Graphics Forum (Proc. Eurographics)*, 2018. 5, 7

[17] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 2

[18] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NIPS*, pages 365–376, 2017. 2

[19] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *CVPR*, pages 3907–16, 2018. 2, 3

[20] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29, 2013. 2

[21] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *AMI*, 16(2):150–62, 1994. 4

[22] Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *ECCV*, volume 8695, pages 154–69, 2014. 2

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 4

[24] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3D data. In *CVPR*, pages 5648–5656, 2016. 1, 2

[25] Danilo Jimenez Rezende, SM Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3D structure from images. In *NIPS*, pages 4996–5004, 2016. 2

[26] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988. 1

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[28] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *CVPR*, pages 2897–2905, 2018. 2

[29] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017. 2

[30] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *IROS*, pages 2442–2447. IEEE, 2017. 2

[31] Hanqing Wang, Jiaolong Yang, Wei Liang, and Xin Tong. Deep single-view 3D object reconstruction with visual hull embedding. *arXiv:1809.03451*, 2018. 1, 2

[32] Weiyue Wang, Qiangui Huang, Suya You, Chao Yang, and Ulrich Neumann. Shape inpainting using 3d generative adversarial network and recurrent convolutional networks. *arXiv:1711.06375*, 2017. 1, 2

[33] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[34] Eric H Warmington, Philip G Rouse, and WHD Rouse. *Great dialogues of Plato*. New American Library, 1956. 1

[35] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. MarrNet: 3D shape reconstruction via 2.5D sketches. In *NIPS*, pages 540–550, 2017. 2

[36] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *NIPS*, pages 82–90, 2016. 1, 2, 3, 5, 6

[37] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D Shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–20, 2015. 1, 2

[38] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *NIPS*, pages 1696–1704, 2016. 3, 5, 6

[39] Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trigoni. 3d object reconstruction from a single depth view with adversarial learning. *arXiv preprint arXiv:1708.07969*, 2017. 2

[40] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 1, 3