Front2Back: Single View 3D Shape Reconstruction via Front to Back Prediction

Yuan Yao¹  Nico Schertler¹  Enrique Rosales¹,² Helge Rhodin¹  Leonid Sigal¹,³ Alla Sheffer¹
¹University of British Columbia ²Universidad Panamericana ³Canada CIFAR AI Chair, Vector Institute
{rozentil, nschertl, albertr, rhodin, lsigal, sheffa}@cs.ubc.ca

Abstract

Reconstruction of a 3D shape from a single 2D image is a classical computer vision problem, whose difficulty stems from the inherent ambiguity of recovering occluded or only partially observed surfaces. Recent methods address this challenge through the use of largely unstructured neural networks that effectively distill conditional mapping and priors over 3D shape. In this work, we induce structure and geometric constraints by leveraging three core observations: (1) the surface of most everyday objects is often almost entirely exposed from pairs of typical opposite views; (2) everyday objects often exhibit global reflective symmetries which can be accurately predicted from single views; (3) opposite orthographic views of a 3D shape share consistent silhouettes. Following these observations, we first predict orthogonal 2.5D visible surface maps (depth, normal and silhouette) from perspective 2D images, and detect global reflective symmetries in this data; second, we predict the back facing depth and normal maps using as input the front maps and, when available, the symmetric reflections of these maps; and finally, we reconstruct a 3D mesh from the union of these maps using a surface reconstruction method. Our experiments demonstrate that our framework outperforms state-of-the-art approaches for 3D shape reconstructions from 2D and 2.5D data in terms of input fidelity and details preservation. Specifically, we achieve 12% better performance on average in ShapeNet dataset [3], and up to 19% for certain classes of objects (e.g., chairs and vessels).

1. Introduction

Humans are amazingly adept at predicting the shape of 3D objects from a single 2D image, a task made particularly challenging by the need to predict occluded surface geometry. Early computer vision researchers hypothesized that humans achieve this goal by employing a series of intermediate representations, of progressively increasing complexity, and envision the 2.5D visible surface geometry as one of these representations [26]. This argument motivates multi-step 3D reconstruction pipelines that use 2.5D visible surface as an intermediate representation [52]. Our novel Front2Back algorithm targets complete 3D surface reconstruction from such an intermediate representation, encoded via multiple depth and normal maps that capture global reflective symmetries (Figure 1b, 1c). We create a complete image to 3D reconstruction framework by combining Front2Back with a data-driven predictor that computes accurate 2.5D visible surface geometry from raw 2D images undoing perspective effects (Figure 1a).

Recent methods directly learn 3D shape representation from raw 2D images [6, 7, 16, 27, 43, 49] or intermediate 2.5D visible surface maps [39, 48, 50, 52] and represent 3D shapes using voxelsizations or implicit functions (defined over \( \mathbb{R}^3 \)). Such models often implicitly distill shape prior information into the network through a large set of 2D-3D paired exemplars obtained from ShapeNet [3] or similar large scale 3D model repository. These methods, collectively, produced an increasingly impressive set of results on single-view 3D shape reconstruction, but are limited in...
terms of fidelity, a limitation which can be largely attributed to the resolution of the resulting voxel grid, as well as to the inherent complexity of learning volumetric ($\mathbb{R}^3$) information directly from images ($\mathbb{R}^2$). While the dimensionality challenge can be sidestepped via the use of 2D (texture) atlases describing the output point-clouds [16] or the use of deformable templates [44, 47], these approaches face other challenges. Notably, none of these methods take explicit advantage of geometric cues correlating visible and occluded surface parts; nor leverage strong perceptual cues in the input image, such as symmetry, which has been shown to be integral for the human 3D perception [32].

Similar to [16] we utilize an image-centric 2.5D intermediate data representation which allows us to describe output point-clouds precisely. However, contrary to all the above approaches we explicitly leverage geometric and perceptual linkages between visible and occluded surface geometry. The core observation behind our framework is that most everyday objects can be almost entirely described by pairs of oppositely oriented height-fields; moreover this property holds for most orientation choices. Therefore, a pair of 2.5D visible surface images taken from opposite views frequently describes a (nearly) complete 3D model (Figure 1b, 1d). Notably for orthographic projection, the silhouettes of such opposite images always align. Consequently, the problem of reconstructing 3D models from single, non-accidental, 2.5 orthographic visible surface images can for many classes of objects be effectively reduced to the Front2Back problem: correctly recovering the back facing 2.5D visible surface given a front facing one. The core advantage of this approach is that instead of directly obtaining 3D data from a 2D or a 2.5D input, a process that is still not well understood, we can now recast our core problem as synthesizing one type of, silhouette aligned, image (back) from another (front), a task that had been successfully tackled by recent image-to-image translation networks. Lastly, and critically, many everyday shapes exhibit global reflective symmetries, which can be accurately detected from front visible surface maps alone. These symmetries allow viewers to predict backfacing geometry as well as to complete details not available in either front or back views.

Following these observations, Front2Back employs an image-to-image translation network to predict back view 2.5D visible surface maps (depth+normal) from the input front 2.5D representation. It then reconstructs a 3D surface mesh fusing together the front and back geometry information using a reconstruction framework capable of producing watertight surfaces by combining positional and normal information [21]. In both the prediction of the intermediate 2.5D back view maps and the reconstruction itself, it leverages global reflective symmetries, algorithmically detected on the front visible surfaces, when present: reflected front surface maps (Figure 1c) are provided as auxiliary input for back prediction, and used as an additional source of data for subsequent reconstruction. To achieve direct 3D reconstruction from raw images we combine our method with a learning based framework that, for each input image, computes the corresponding orthographic 2.5D visible surfaces rectifying perspective distortion present in these images. Our approach leads to significant improvements in the resulting 3D shape reconstruction, compared to competing state-of-the-art methods, resulting in 12% improvements on average, measured using mesh-to-mesh distance [8] and up to 19% improvements on certain ShapeNet object classes, compared to the closest competitors.

**Contribution:** Our core contribution is a novel framework for 2.5D visible surface map to 3D reconstruction, which combines recent learning-based approaches with more traditional geometric processing and produces results superior to the state-of-the-art. This contribution is made possible by two core technical innovations: (1) the use of 2.5D back maps predicted from the 2.5D front maps as a stepping stone toward full 3D reconstruction, and (2) the use of symmetries detected in the front view to significantly improve the performance of both back prediction and 3D reconstruction.

**2. Related Work**

**Single-view 3D Surface Reconstruction.** Reconstructing 3D models from single view 2D images, or 2.5D depth (and normal) maps, is a difficult and ill-posed problem. Recent learning based methods show promising results in addressing this challenge [40]. While many methods predict 3D shape representations directly from 2D images, e.g., [6, 7, 11, 13, 16, 27, 43, 44, 49, 51], others, e.g., [39, 48, 50] first reconstruct the 2.5D visible surface (typically representing it via depth and normal maps) and then use this intermediate representation as a stepping stone toward complete 3D reconstruction. Many methods in both categories represent the reconstructed shapes using voxels, e.g., [13, 48, 50, 51] or limited-depth octrees [45]. The accuracy of the reconstructions produced by these methods is limited by the finite resolution of the voxel or octree cells, limiting the methods’ ability to capture fine details. Template based methods [37, 44, 47] perform well when the topology of the input template matches that of the depicted shape, but are less suited for cases where the target topology is not known a priori. Recent implicit surface based methods [6, 27, 34] strive for resolution independence but require watertight meshes for training. Since the vast majority of meshes in the wild are far from watertight instead of training directly on this data they use watertight approximations which necessarily deviate from the originals. This de-

---

1 A randomly tested sample of 520 ShapeNet core shapes had on average 80% of their surface visible from 95% of random opposite view pairs.

2 97% of the models in the test split of ShapeNet core are not watertight.
viation can potentially bias surface prediction. Atlas based
reconstruction avoids these pitfalls, but exhibits similar
accuracy levels. Point or depth-only maps based methods
produce collections of points close to the target objects surface; however surface reconstruction from unori-
ented points is a challenging problem in its own right [2],
thus when attempting to recover the surface from cloud the
output quality decreases dramatically [36]. Our method
uses depth plus normal maps as intermediate representation,
works well on shapes with diverse topology, has no bounds
on accuracy beyond the input image resolution; and can be
directly trained on models with arbitrary connectivity and
non-manifold artifacts.

Some methods used opposite views for reconstruction.
Matryoshka Networks [33] predicts three pairs of depth
maps in axis-aligned opposite views. Two recent methods
reconstruct front and back depth map images of human sub-
jects from photographs [12, 29]. Our method differs from
those in utilizing a normal map alongside depth, symmet-
ric cues, and normalized, orthographic coordinates obtained
through perspective correction to predict depth maps across
a wide range of classes/views.

Symmetry priors were used in earlier works [22, 30, 38,
41, 42] for completing partial scans or estimating depth of
symmetric points. Recently, [47] used symmetry to regular-
ze out ops at training time. In contrast, we integrate sym-
metry as a cue, in both training and test, for reconstruction.

**View Synthesis and Shape Completion.** Our core task of
predicting back view 2.5D surface can be seen as a special
case of alternative view synthesis [9, 10, 15, 25, 53]. Most
such methods aim to predict views that have similar view-
points to the original. Recent approaches, [5] render comp-
elling images and depth maps from viewpoints that differ
by up to 40° from the original. Surface reconstruction from
depth maps alone [24, 36] suffers from similar pitfalls as
one from unoriented clouds [2]. In contrast to these settings
we seek to recover the exact opposite view, where the only
overlap between the observed surfaces is along the outer
silhouette. By combining these strategically selected views
and utilizing symmetry we successfully compute oriented
point-clouds, allowing for the use of more robust recon-
struction methods [21]; producing superior reconstructions.

Learning based shape completion methods, *e.g.,* [17, 31],
attempt to extend partial surfaces across poorly captured re-
regions to produce a complete shape representation. These
methods are designed to operate on 3D shape inputs, are
typically limited to filling relatively small holes, and em-
ploy shape priors for ambiguity resolution. In contrast, our
Front2Back step predicts a complete surface from only front
facing depth+normal maps and uses an intermediate 2.5D
back map to achieve this goal.

**Image-to-image translation.** Image-to-image translation
[19, 20, 23, 46, 54] is a powerful tool for synthesizing new
images from existing ones for applications such as image-
from-sketch synthesis [35] and make-up application [4].
While typical translation methods aim to preserve the view
and the content of the original image and only change some
of their visual attributes, generating back views from front
ones requires significant changes in the depth and normal
content, a much more challenging task.

3. Approach

Our method takes as input a single perspective 2D image
and generates a 3D mesh of the corresponding object, using
four key steps (Figure 2). We start the process by predicting
orthographic silhouette, depth, and normal maps of the por-
tion of the target object’s surface visible in the input image
(Section 3.1). We proceed to locate a global 3D reflective
symmetry plane from this visible surface, if one exists (Sec-
tion 3.2). We use the located plane (if detected) to infer the
occulted parts of the shape whose symmetric counterparts
are visible, by reflecting the input maps over the symmetry
plane to get a second set of depth and normal maps (hereon
referred to as reflected maps). We mask all depth and nor-
mal maps using the silhouette, denoting all pixels outside as
background, and use these maps as input to our core back
prediction stage. The prediction stage takes this input and
generates new depth and normal maps for the back view,
the exact opposite of the input front view (Section 3.3).
We perform this prediction using a variant of conditional gen-
erative adversarial networks for image to image translation.
Finally, we combine the front view maps, reflected maps,
and predicted back view maps to extract the corresponding
oriented point cloud and reconstruct the surface from this
cloud (see Section 3.4 for details).

3.1. Orthographic Front View Prediction

For the 2D to 2.5D step, we adopt and train the 2.5D es-
timation network of [50], using example input-output pairs
of perspective images and corresponding same view direc-
tion orthographic depth, normal, and silhouette maps. Per-
spective rectification simplifies subsequent symmetry esti-
mation and allows us to enforce the same silhouette con-
straint across all computed maps.

We define the loss function as the sum of the three in-
dividual $L_1$ losses of the outputs. The original network is
designed for noisy, real-life images and purposefully adds
noise to input data to mimic real-life artifacts; since similar
to most recent single view reconstruction methods our train-
ing image set is synthetic, for a fair comparison we disabled
this feature in our implementation.
3.2. Symmetry Detection

Reflective symmetry is a frequent feature of both organic and man made shapes. It plays a vital role in human perception - absent information to the contrary, human observers expect symmetries observed in the visible parts of a surface to extend into occluded parts facilitating mental reconstruction of these occluded surfaces [18]. Our method mimics this behavior by explicitly formulating a surface hypothesis derived from reflective symmetries detected on the front view maps. Efficient, robust, and accurate detection of reflective symmetry planes from partial surfaces is a challenging geometry processing problem [1, 28]. While existing methods are designed for dense point-clouds, we seek to detect symmetries on pixelated and thus heavily quantized data, which frequently has very low local resolution (e.g. chair legs which are less than 5 pixels wide). Most critically, we seek to avoid false positives, as inaccurate reflected maps can severely impact our subsequent back prediction and reconstruction steps.

We design a targeted two-step reflective symmetry plane detection method, that addresses these challenges by combining information across all three front maps. We first estimate an approximate symmetry plane using a clustering-based approach, sped up using a variant of RANSAC. We refer to this plane as initial plane. We then optimize this initial plane to better align the original oriented points with their reflected counterparts using an iterated closest point method (Figure 3). We avoid false positives in both steps by utilizing two key constraints. We note that the silhouette map defines the visual hull of the target object, thus we expect any surface parts symmetric to parts of the visible surface to lie inside it; we thus filter out reflection planes that produce maps violating this constraint. Since our front maps are expected to be at least as accurate as the reflected ones, we similarly only consider planes that produce reflected maps that do not occlude the front surfaces (i.e., have no points closer to the viewer at the same x-y locations), we refer to this constraint as visibility. We use the detected planes to generate reflected depth and normal maps.

![Algorithm Stages](image)

**Figure 2:** Algorithm Stages: (left to right): input perspective image; 2.5D orthographic map (depth+normal+silhouette) prediction; detection of reflective symmetries (regions on the left and right of the plane colored blue and red respectively); back map prediction; final surface reconstruction.

![Symmetry plane detection](image)

**Figure 3:** Symmetry plane detection. Given front view maps (insets, left) we use plane-space clustering to obtain initial reflective planes (center, dashed) and obtain the final planes (solid, right) by using ICP iterations. (For many inputs the improvement is more subtle.)

**Initial plane.** We use a clustering-based symmetry detection algorithm, inspired by [28] for estimating the initial symmetry plane. We first transform the front-facing normal and depth map into an oriented point set $P$. For each pair of oriented points $(p_i, p_j) \in P \times P$, we calculate the plane that best reflects $p_i$ to $p_j$. We then repeatedly sample subsets of such potential symmetry planes and cluster each subset using mean-shift clustering. We compute a Voronoi diagram of all planes with respect to the centers of the resulting clusters, and define the score of each center as the number of planes in its' respective cell. To select the initial plane among cluster centers obtained across all iterations we first discard all centers that produce reflected maps significantly violating visual hull or visibility constraints. We then select as initial plane the center with the highest score.

**Optimization.** Our initial plane is based on a sampling approach and only considers planes defined directly by point pairs, thus while close to the optimal, it can often be further improved. We optimize the plane using a variant of classical ICP. At each iteration, for each point $p \in P$, we first calculate the point $r_p \in P$ that is closest to its reflection around the current symmetry plane $s$. We prune all correspondences whose distance exceeds a user-definable threshold or whose normals are not reflected within a tolerance, and use the remaining correspondences $P_e \subseteq P$ to optimize for a symmetry plane that maps the points closer to their reflections, i.e.,

$$s' = \arg \min_s \sum_{p \in P_e} \|\rho_s(p) - r_p\|^2. \tag{1}$$
We solve this minimization problem using gradient descent with back tracking line search to determine the step size and update the estimate for the symmetry plane with $s'$. We repeat the process until convergence (Figure 3, right).

**Determining Model Symmetry.** We classify the input as asymmetric if the resulting reflected maps violate visual hull or visibility constraints with tight thresholds, or generates a map which covers less than 40% of the silhouette interior. False symmetry can dramatically impact reconstruction quality, far more so than reconstruction with no symmetry information, motivating us to use fairly strict thresholds. We classify about $\approx 20\%$ of the data as asymmetric.

### 3.3. Back View Maps Prediction

Recent deep learning-based image-to-image translation methods focus on the translation between different domains. In our method, we demonstrate that such deep neural networks can also be used to learn the mapping between 2.5D representations in opposing views.

Our learning model is similar to [20], which is based on a conditional generative adversarial network (cGAN). The architecture consists of two networks: a generator and a discriminator. The discriminator is trained on a training data set to classify an input image into either real or fake w.r.t. training data set. The generator is trained to produce images that the discriminator evaluates as real. Our goal is to predict the back view normal and depth map from front view maps. To integrate symmetry information, we also feed the reflected depth and normal maps into the network when symmetries are detected (see end of Section 3.2). Hence the input to the generator is a depth-4 or -8 image, depending on symmetry circumstance, and the output is a depth-4 image encoding predicted back depth and normal maps. To train the network, we use a loss function $L$ that comprises terms for each of the individual aspects of our prediction problem:

$$L = w_{GAN}L_{GAN} + w_dL_d + w_nL_n.$$  

(2)

The GAN loss $L_{GAN}$ is the traditional loss used for generative adversarial networks that steers the interplay between generator and discriminator. The two similarity loss functions for depth $L_d$ and normals $L_n$ aim to measure the pixel-wise differences between predicted and ground truth maps. In our experiments, we set $w_{GAN} = 1$, $w_n = 100$, and $w_d = 1000$. Next, we present loss functions in more detail.

**Adversarial loss.** We use the traditional adversarial loss as presented in [14]. Given the front view normal and depth maps $\hat{N}_f$ and $\hat{D}_f$, the reflected back view maps $\hat{N}_b$ and $\hat{D}_b$, we define the adversarial loss as:

$$L_{GAN}(G, D) = 
\mathbb{E}_{(N_f, D_f)}[\log(D(\hat{N}_b, \hat{D}_b))] + 
\mathbb{E}_{(N_f, D_f, \hat{N}_b, \hat{D}_b)}[\log(1 - D(G(N_f, D_f, \hat{N}_b, \hat{D}_b)))] .$$  

(3)

**Similarity loss.** Similar to many existing image-to-image translation tasks, we use L1 loss between the output and ground truth images as a similarity measure for the depth maps. Since differences between orientations or normals are more accurately represented by angles, we use cosine similarity for the normal maps. Given the predicted back view maps $\hat{N}_b$ and $\hat{D}_b$ produced by the generator and the ground truth normal and depth images $N_b$ and $D_b$:

$$L_d = \| \hat{D}_b - \hat{D}_b \|_1$$

(4)

$$L_n = \begin{bmatrix} N_b, \hat{N}_b \end{bmatrix}_{\cos},$$

(4)

where

$$[A, B]_{\cos} = \sum_{i,j} \left(1 - \frac{A(i, j) \cdot B(i, j)}{\| A(i, j) \| \cdot \| B(i, j) \|} \right).$$

(5)

### 3.4. Surface Reconstruction

We fuse the per-pixel positions and normals form the front, reflected, and predicted back maps to generate an oriented point cloud and use screened Poisson [21] for surface reconstruction. To produce closed meshes we use Dirichlet boundary conditions, and use an interpolation weight of 4 to promote interpolation of input points. Our fusion process automatically corrects quantization artifacts and inaccuracies in the computed maps that can lead to catastrophic reconstruction failures, and accounts for point density assumptions made by typical reconstruction methods.

**Fusion.** Similar to many other methods, screened Poisson reconstruction [21] expects input points originating from the same surface to be closer to one another than to points originating from an oppositely oriented surface. To satisfy this property given any pair of map points with oppositely oriented normals (one front facing and one back facing) which are within image-space distance of less than two pixels of one another, we ensure that the depth difference between them is at least two pixels, by moving the back facing point backward if needed. Following the visibility prior, we expect the front map points to be more reliable and closer to the viewer than points from other sources, thus we discard reflected and back map points if they are closer to the viewer than front map points at the same $(x, y)$ position. Following the same argument, we expect back map points to be farthest from the viewer; however, in cases of depth conflicts we trust reflected maps more that back prediction. Thus we discard back map points that are closer to the viewer than reflected map points at the same $(x, y)$ position. Lastly, we remove outliers which we classify as points which are above a fixed threshold away from all four of their immediate image-space neighbors in the same map along the depth axis. The logic behind this criterion is that while we expect the depth maps to exhibit discontinuities, we do expect local features to be larger than one pixel.

More fine grain implementation details are provided in the supplementary material.
4. Experiments

We tested our methods on a large body of models across different classes and performed both qualitative and quantitative comparisons, as well as ablation studies demonstrating the impact of the different algorithmic choices made. Additional results, including ones generated from 256 × 256 images, are provided in the supplementary material.

Dataset. We use the ShapeNet Core dataset [3] and its official training/testing split, which includes 13 object categories. To compare against both methods that use this split and those that use the training/testing split of Choi et al. [7], we report all quantities on the intersection of the two test sets that none of the models have seen in training. To be comparable, we train our orthographic front view prediction using the training split of the images corresponding to the training models provided by Choi et al. [7] at resolution 137 × 137 pixel. We follow [27] and others and use the first, random view, provided by Choi et al. for the test set. To generate both the front and back view maps for training, we rendered orthographic depth and normal map from these views and their opposites. We use the canonical symmetry plane (yz) of the training shapes that are symmetric around it for generating reflected maps for training.

Metrics. For evaluation we use the mesh-to-mesh symmetric $L_1$ distance (MD) [8] and Chamfer $L_1$ distance (CD) between the ground truth and reconstructed meshes. We measure MD using Metro [8], a well established measurement tool in the geometry processing community, using default Metro parameters. We measure CD using the implementation provided by [27] across 100K evenly sampled points on the two meshes. The core difference between these metrics is that Metro looks for closest distance from a sample point on one mesh to any of the triangles on the other, while CD only considers sample to sample distances and thus is inherently more dependent on sampling density / quality. We report MD using the diagonal of the ground truth bounding box as unit 1, and follow [27] for reporting CD.

Implementation Details. We use ‘adam’ optimizer with learning rate 0.0001 and batch size of 4 for training our orthographic front view predictor. We use the loss and corresponding architecture and parameter settings from [50]. For symmetry detection we use 20 iterations each with 8K plane samples when computing initial plane up to 400 iterations for ICP; for correspondence pruning we use thresholds of 4 pixels and 60° on position and normals. We reject reflection planes as violating the visual hull threshold, if over 5% of the reflected pixels are at least 5 pixels outside the silhouette, and reject them as violating visibility if over 15% of the pixels are in front of the front view. We use ‘adam’ optimizer with learning rate 0.0002 and batch size of 1 for training our back prediction network, and use 5 random views out of the 24 provided by [7]. For final reconstruction step outlier removal we use threshold of 4. These parameters are kept constant for all experiments.

Figure 4: Back map prediction. Given front and reflected depth+normal maps our image-to-image translation network predict the corresponding back view maps; we correctly identify asymmetric models and predict the corresponding back maps using a separately trained predictor which uses only front maps as input.

4.1. Assessing Back Map Prediction

Figure 4 shows the results of our core back view from front view prediction step for several different-class examples. The depth and normal maps clearly convey that our networks can successfully predict the backside of the raw input images, replicating front features when appropriate and consistently completing entirely invisible features such as the back of the sofa or the occluded chair legs.

Back Prediction Ablation. To verify that the symmetry information improves back map prediction, we tested our prediction method with and without the reflected maps. We measure this impact on ground truth front maps of models from the test set to decouple performance of back prediction from the rest of the method. Table 2 shows the impact of using reflected maps on the accuracy of the back prediction.
for airplanes. We report performance in terms of average \( L_1 \) distance for the depth and average one minus cosine distance for the normal map; effectively this results in Eq. (2).

As the numbers consistently show the use of reflection maps during prediction is important for the accuracy of resulting back maps. Figure 4, illustrates this difference on real image data. The impact of symmetry is most apparent in the normal maps, where chair legs are easily recognizable.

![3D shape from real images. Our results (right) produced from real images (left) While not perfect, they provide a reasonable approximation of the shown shape.](image)

Table 1: Surface ablation study on airplane models, measuring mesh-to-mesh distance to ground truth on surfacing outputs generated without (top) and with (bottom) reflection maps. Our outputs are closer to ground truth and visually more symmetric (Figure 5).

<table>
<thead>
<tr>
<th>Ours (no symmetry)</th>
<th>MD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00332</td>
<td></td>
</tr>
<tr>
<td>Ours (with symmetry)</td>
<td>0.0029</td>
</tr>
</tbody>
</table>

Table 2: Front map prediction ablation: on airplanes. Ground truth front maps are used with (bottom row) and without (top row) reflected maps. Performance is reported in terms of average per pixel and across model \( L_1 \) and 1 minus cosine distance of depth and normal back predictions respectively. Clearly, the reflected maps are beneficial.

<table>
<thead>
<tr>
<th></th>
<th>depth (avg ( L_1 ))</th>
<th>normal (avg ( 1 - \cos(a', a) ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (no sym)</td>
<td>0.000593</td>
<td>0.00310</td>
</tr>
<tr>
<td>Ours (with)</td>
<td>0.000878</td>
<td>0.00268</td>
</tr>
</tbody>
</table>

3D Reconstruction Ablation. The final surface reconstruction uses as input a union of points from the front, reflected front, and predicted back views. To evaluate the importance of using these maps for reconstruction, we perform an ablation study on chairs and airplanes where we only feed some of the available maps into the Poisson reconstruction (Figure 5). Somewhat self-evidently, accurate reconstruction is essentially impossible absent back view information. More interestingly, our measurements (Table 1) show that including the reflected maps in the point cloud used for final reconstruction makes a more subtle but important impact on the quality of the results. The qualitative impact of incorporating the reflected map points is also quite significant (see Figure 5) - human observers expect many man-made objects to be symmetric, and reconstructions that do not respect symmetry appear far less satisfactory from a user perspective. The results incorporating symmetry are thus both better quantitatively and more visually realistic / believable.

4.2. Single view 3D Shape Reconstruction

We tested our method on both 137 and 256 resolution images from the ShapeNet dataset [3] as discussed above. Representative results are shown in Figures 1, 2, 7, and 6. Our method generates high quality results across a large number of classes and object geometries.

Comparisons. We compare our method to a range of state-of-the-art techniques, including Pixel2Mesh [44], AtlasNet [16], OccNet [27], and IM-NET [6] (Table 3). We use published codes and pre-trained weights from respective papers to reproduce all results. We are unable to provide direct comparison to 3DN [47] because of issues with their published code. Figure 7 shows some representative comparisons. As demonstrated across multiple inputs our method consistently captures fine model details more accurately than other methods. We note that while we use the CD metric from [27] to report the performance, we are recomputing all the numbers. Both [27] and Chen et al. [6] rely on watertight training models, that only approximate ground truth, and report Chanfer distance wrt to these proxies, while we seek to measure distances wrt to the real ground truth. Table 3 verifies quantitatively what the preceding qualitative results have shown: Our reconstructions are much closer to the ground truth shapes than those of other learning-based shape reconstruction methods, reducing both MD and CD by as much as 19% (on cars) and 18% (on lamps) respectively. On average across the 13 categories we are 12.5% better than the closest competing method, which turns out to be [44]. Overall, our method is more accurate on 9 out of 13 categories; performing worse on cars, cabinets, phones and sofa where it is only marginally worse than the best method.

Challenging Inputs. We measured front2back performance on inputs that challenge our core assumptions about visibility and symmetry. Approximately 25% of the views in our test set [7] jointly with their opposite views reveal less than 80% of the ground truth model’s surface. Table 4 shows that our method continues to outperform prior methods on these inputs, achieving an average reconstruction error (MD) of 0.0165. Notably we perform much better on the rest of the inputs, where the combined visibility is higher (\( MD = 0.014 \)). This robustness to drop in visibility is due to the use of symmetry cues during reconstruction (Table 1): using the reflected front point cloud (Figure 5), enables us to recover geometry occluded in both front and back views, whose reflection is present in the front view.
Figure 7: Qualitative comparison to state-of-the-art. Visual comparison of our results and those produced by Pixel2Mesh [44], AtlasNet [16], OccNet [27], and IM-NET [6]. In all examples our results are more consistent with the input images. For some of the methods, e.g. [27, 6] the strong shape priors result in meshes that are close to what the network considers a reasonable object but very far away from the input image.

Table 3: Comparisons against state-of-the-art. We compare our results against Pixel2Mesh [44], AtlasNet [16], OccNet [27], and IM-NET [6] measuring both mesh-to-mesh distance (MD) and $L_1$ Chamfer Distance (CD). Our method provides the best results overall for both metrics, outperforming the closest competitors on nine out of 13 classes.

Table 4: Mesh-to-mesh distance (MD) on challenging inputs (less revealing views or non-symmetric models).

of inputs in our tests (e.g., sofa in Figure 4) were classified as non-symmetric at runtime, and had the rest of the computation (back prediction + reconstruction) performed without using symmetry information. Our average reconstruction error (MD) across these inputs was 0.0164 compared to 0.0137 on inputs where symmetry was detected and utilized. Even on this non-symmetric subset our error is lower than those of the other methods as shown in Table 4. These measurements confirm that our method is not limited to symmetric inputs or inputs where front and (predicted) back views jointly reveal the vast majority of the observed object’s surface.

Generalization. We tested our front2back chair model (generating back 2.5D maps from front ones) on the bench and sofa classes, and tested the bench model on the sofa class. In all three settings, the increase in average error was 0.001 compared to using the dedicated models. We also trained a front2back model on the union of all training sets. The average reconstruction error using this model was 0.0152 compared to 0.0144 for our individual models. In all tests we keep the dedicated per-class image2front models; this illustrate generalization of our core front2back method.

Application to real images. Figure 6 shows that, although trained on synthetic renderings, our method yields realistic reconstructions on real images when provided with a segmentation mask or simple background.

5. Conclusions

We presented a novel single view 3D reconstruction method anchored around prediction of back view maps from front view information. Key to the success of our method, which shows improvements over SoTA, is a combination of advanced learning approaches with geometry priors that motivate our algorithmic choices. Notably, our prediction of 2.5D front maps from images can be potentially substituted by other sources of depth and normal maps, such as depth scanners or sketch processing systems, enabling direct reconstruction of complete 3D objects from such data.

Acknowledgments

The authors wish to thank the reviewers for their insightful suggestions and NSERC for funding support.
References


[29] Ryota Natsume, Shunshke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeho Morishima. Siclope: