

X-Section: Cross-section Prediction for Enhanced RGB-D Fusion

Andrea Nicastro¹, Ronald Clark¹, Stefan Leutenegger²

¹Dyson Robotics Lab, ²Smart Robotics Lab, Imperial College London

a.nicastro15@imperial.ac.uk

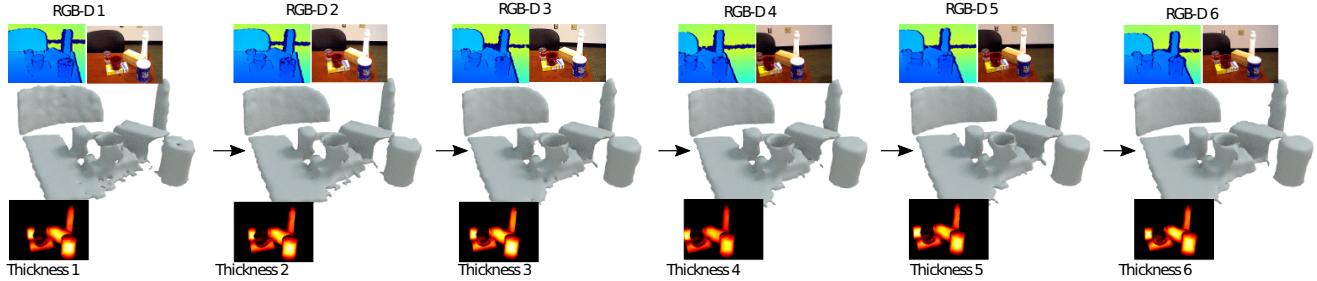


Figure 1: Our approach uses predictions of the objects cross-sectional thickness to improve volumetric reconstruction quality. Top row shows the input to the proposed pipeline, an RGB-D frame. Bottom, cross-section prediction. From left to right in the middle, incremental reconstruction via our enhanced TSDF fusion algorithm.

Abstract

Detailed 3D reconstruction is an important challenge with application to robotics, augmented and virtual reality, which has seen impressive progress throughout the past years. Advancements were driven by the availability of depth cameras (RGB-D), as well as increased compute power, e.g. in the form of GPUs – but also thanks to inclusion of machine learning in the process. Here, we propose X-Section, an RGB-D 3D reconstruction approach that leverages deep learning to make object-level predictions about thicknesses that can be readily integrated into a volumetric multi-view fusion process, where we propose an extension to the popular KinectFusion approach. In essence, our method allows to complete shapes in general indoor scenes behind what is sensed by the RGB-D camera, which may be crucial e.g. for robotic manipulation tasks or efficient scene exploration. Predicting object thicknesses rather than volumes allows us to work with comparably high spatial resolution without exploding memory and training data requirements on the employed Convolutional Neural Networks. In a series of qualitative and quantitative evaluations, we demonstrate how we accurately predict object thickness and reconstruct general 3D scenes containing multiple objects.

1. Introduction

Knowledge of the shape of objects and of unseen part of the scene plays a critical role in applications such as robotic manipulation and autonomous exploration. In robot manipulation, the understanding of object geometry clearly influences the choice of grasping points. Similarly, in autonomous navigation, any additional information about occupied versus free space in the scene is helpful. The fusion of unseen information in the mapping process leads to more efficient exploration and faster map coverage.

Recent advancements in machine learning have fuelled improvements in single view 3D reconstruction. However, the developed techniques are not necessarily readily integrated with state of the art spatial mapping systems.

In this work, we propose a novel approach to object reconstruction embedded in a scene that allows scalable multi-view reconstruction of both individual objects and groups thereof. The task we propose is to predict the geometry behind sensed surfaces in the form of view-centred cross-sectional thickness. We embed the thickness prediction network, X-Section, in a pipeline that allows to scale our approach to scene level. To integrate multiple views and recover 3D geometry, we suggest a modification to truncated signed distance function (TSDF) fusion. Furthermore, our framework can be easily paired with other mapping approaches such as Bayesian probabilistic mapping [23].

There are several reasons to prefer 2D predictions rather

than trying to estimate the full 3D shape in one shot. One of the main advantages is that predicting an image instead of a voxel grid avoids the explosion in the number of weights of the network. Moreover, the use of a reconstruction algorithm to recover 3D geometry loosens the coupling between the reconstruction resolution and the network prediction. In an extensive study of different types of learning-based reconstruction approaches [32], the authors also found that view-centred pixel-wise predictions generalise better to unseen classes than object-centred voxel-based models.

As obtaining training data for this task is challenging, we introduce a new dataset consisting of both synthetic and real images. We render RGB, depth and thickness for models of the YCB Dataset [3] with domain randomisation. To achieve good performance on real data we fine-tuned on real sequences from [43] with rendered thickness of the aligned objects. Similar to an X-Ray machine we render thickness by raytracing through synthetic models of objects and measuring the distance between the observed surface and the first surface behind it. An illustration of this cross-sectional thickness is shown in Figure 2.

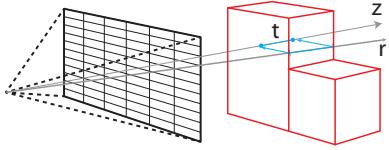


Figure 2: Illustration of the cross-sectional thickness. t is the thickness of the surfaces hit by the ray r and projected on the principal axis Z .

In short, we claim the contribution of our work to be fourfold:

- A novel task to predict view-dependent 2D per-pixel thickness that can be used to efficiently recover a 3D volume.
- A complete pipeline from RGB-D or depth and silhouette (DS) to a full 3D reconstruction for 3D tabletop scenes using predicted thickness.
- A dataset of thickness data for 106k synthetic plus 34k real views of YCB objects, along with the RGB, depth and silhouette images and the code to render more views.
- Training and prediction code with pre-trained weights to reproduce results.

The structure of the paper is as follows. We first review related works on volumetric fusion, RGB-D shape completion and some single view RGB reconstruction approaches.

We then introduce our approach and the dataset we train our model on. Finally we evaluate our model’s performance on real RGB-D sequences.

2. Related work

Surface Prediction and Spatial Mapping The most popular approach for reconstructing scenes from RGB-D images involves registering and fusing multiple frames into a 3D voxel grid. This volumetric fusion approach, popularised by KinectFusion [27], works by first tracking the camera pose and then it uses the integration approach of Curless and Levoy [9] to fuse the depth images into the volume. Various improvements have been introduced, mainly focused on reducing tracking drift [7] and increasing the size of scenes that can be reconstructed. Kintinuous [41], for example, uses a sliding volume to map large spaces. BundleFusion [10] reduces tracking drift by global bundle-adjustment and re-integration into the mapping process. [39] tackles the efficiency bottleneck by means of a tree data structure. With the advent of deep learning there has been much interest in learning geometrical, structural and semantic priors to enhance the reconstruction process. For example, [40] makes use of surface normal predictions to improve a monocular reconstruction. [35] uses semantic segmentation along with RGB-D reconstruction to create annotated maps of indoor scenes. More recently, Fusion++ [24] introduced an object-centric approach to large scale mapping which builds a map consisting of multiple TSDFs, each representing a single object instance.

Volume Completion A number of approaches propose to complete the scene starting from RGB-D information. Song et al. [34] and ScanComplete [11] infer the missing voxels in a grid map along with the semantic labels. OctNetFusion [30] describes a deep learnt fusion process using an octree data structure for efficiency. Their scheme can be seen as learning an implicit surface from the depth maps, helping with noise reduction and outlier suppression when fusing. Voxlets [12] operates on partially reconstructed 3D voxel grids. Other approaches [44] use GANs to train an RGB-D to voxel predictor. The main disadvantage of these approaches is that it is inefficient for fusing multiple views as its 3D convolutions are both memory and compute intensive, restricting their use in real time applications.

Silhouette based reconstruction Shape-from-silhouette methods reconstruct the 3D shape of an object using multiple silhouette images taken from different viewpoints [1].

More closely related to our approach is [29], where the authors extract curves along the silhouette and reconstruct the object by finding the smooth surface which adheres to the edge curves. This method, however, requires that the object is symmetric and that the silhouette image is taken perpendicular to the symmetry axis.

Single-view 3D reconstruction Classical approaches to

single-view reconstruction [28, 8, 18, 19, 45] relied on strong geometric priors. While these methods showed some impressive results on simple scenes, they lack ability to capture the complexity of real object shapes.

The advent of deep learning has led to a major boost in the complexity and quality of scenes and objects that can be reconstructed from a single view. Approaches like [6, 31, 38, 20, 14, 42, 13, 46, 2, 15, 38] all attempt to reconstruct 3D objects from 2D views and/or silhouettes. In the best case, these methods provide a view-centred reconstruction requiring to recover the translation and scale of the object, a challenging task itself. In the case the prediction is in a canonical pose, the full pose and scale has to be estimated.

In a concurrent work, [33] represents an indoor scene as four layers of depth. Apart from the first, the layers of depth represent the full extension of an object along the ray. This might create artefacts in the case of non-convex shapes. Our work differs in the definition of the thickness as the distance between the observed surface and it's back and compensate for the incomplete representation of the geometry by means of integration with a multi-frame depth fusion algorithm.

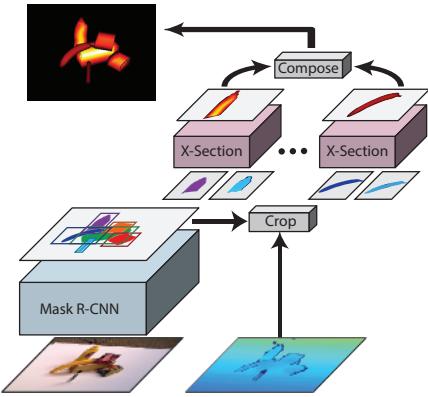


Figure 3: Overview of our cross-section prediction pipeline. An RGB frame is passed to Mask R-CNN. The resulting bounding boxes and masks are used to process RGB and depth data and crop single objects. X-Section is run for every object and the outputs are composed in a thickness frame.

3. Approach

Predicting the thickness for an entire scene is a very demanding problem. Our method is based on the idea that decomposing this complex problem into smaller and simpler tasks makes the solution easier to find. We first decompose the scene into object instances and then produce an estimate for every object in the image. We then compose multiple predictions into a single frame that can be used in the fusion process to obtain a 3D model of the scene.

As can be seen in Figure 3, our system consists of five steps in total. An object detector, a pre-processing stage, a prediction operation and a final composition followed by a fusion step.

First, an object detector takes as input an RGB frame and outputs a set of bounding boxes and masks – we use an off-the-shelf solution for this. At the second stage of the pipeline, the output of the object detector is pre-processed to be input to our estimation network. The X-Section network is run for every object. Finally, the per-object predictions are merged in a single thickness frame and passed to the reconstruction algorithm that outputs a representation of the volume in a voxel grid.

3.1. Object Detection and Instance Segmentation

Our approach relies on any object detector that provides bounding boxes along with a segmentation masks of the object. For the current work, we chose an off-the-shelf version of Mask R-CNN [16] based on ResNet [17] and trained on the MS-COCO dataset [22]. Alternatives to Mask R-CNN include MaskLab [5] or DCAN [4].

3.2. Pre-processing

The output of the object detector has to be pre-processed before moving to the estimation stage. We expand the bounding boxes to have a 4:3 shape ratio and use them to obtain RGB and depth patches along with corresponding silhouettes. To bridge the gap between the training and test depth images, we subtract the mean of the object region and the mean of the background to the corresponding pixels. In this way we aim to push the network to focus only on the shapes rather than on the absolute depth values. Images from a depth sensor are typically incomplete. At test time, we run an additional inpainting step, described in [37], to recover missing data due to sensor noise.

3.3. Thickness Network Architecture

The network we propose to estimate thickness has an encoder-decoder structure in which input images are reduced to a code of dimension 3x4 with 2048 channels. Considering the affinity of our task with object recognition and given the limited size of the available dataset, we use an encoder based on ResNet with pre-trained weights on ImageNet. Since our input differs from the original one the network was trained with, we add an additional convolutional layer that takes stacked depth and silhouette images (or RGB and depth) and outputs a 3 channel feature image. The decoder consists of blocks of upsampling followed by two convolutional layers with ReLu [25] activation along all the layers except for the last one, which is linear. There are no skip connections between the encoder and the decoder part of the network. We train by minimising the \mathcal{L}_2 loss between the predicted and the ground truth thickness. Figure

4 depicts an example architecture based on ResNet101.

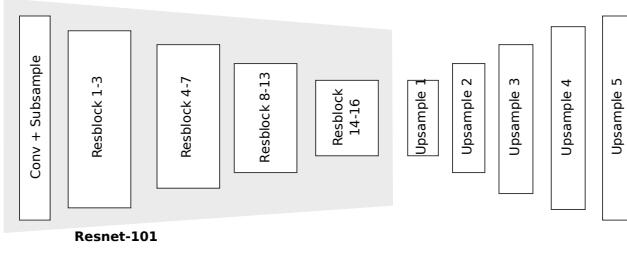


Figure 4: X-Section consists of a ResNet encoder and 5 upsampling blocks. The first layer blends the input in a 3-channel stack used by the encoder. Each upsampling block is composed from *bilinear upsample - conv1 - conv2*. We use no skip layers apart from the residual connections in the encoder.

3.4. Enhanced TSDF Fusion

2D thickness prediction can be used to recover the 3D shape by fusing multiple frames, or even form a single view. To do so, we introduce an enhanced 3D fusion algorithm based on the approach of Curless and Levoy [9]. The affinity of the thickness signal to depth measurement allows for easy integration into existing frameworks.

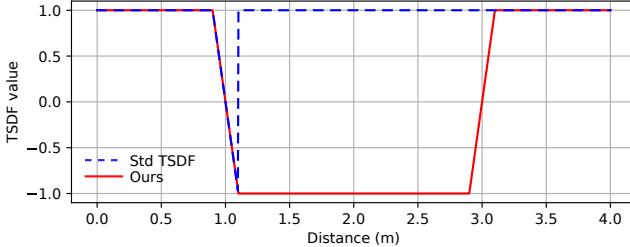


Figure 5: A plot of our thickness enhanced TSDF and standard TSDF. We show an example of surface at 1.0m with thickness 2.0m.

The value of the new TSDF $\phi(z)$ depends on the truncation value τ that define the margins in which front and back surfaces lie respectively; d and t denote the depth and thickness value at a pixel \mathbf{u} and z the position along the ray of the camera corresponding to that pixel :

$$\phi(z) = \begin{cases} 1 & z \leq d - \tau, \\ \frac{d-z}{\tau} & d - \tau < z < d + \tau, \\ -1 & d + \tau \leq z \leq d + t - \tau, \\ \frac{d+t-z}{-\tau} & d + t - \tau < z < d + t + \tau, \\ 1 & z \geq d + t + \tau. \end{cases} \quad (1)$$

The resulting TSDF profile is shown in Figure 5. In contrast to methods such as [27] this reconstruction algorithm

does not only yield surfaces, but explicitly reconstruct the occupied volume of an object. Multiple frames are fused by weighted average of the TSDF for each frame. When a voxel is updated the corresponding weight is incremented.

4. Dataset

In order to generate thickness data we need a dataset with a complete model of each object. Most large-scale RGB-D datasets [26, 34, 21] provide 2D images with depth and object instances but do not provide full 3D data about the objects. A dataset that satisfies this requirement is the YCB dataset [3]. YCB is composed of 92 objects belonging to 77 classes. The dataset provides water tight meshes with textures extracted from images.

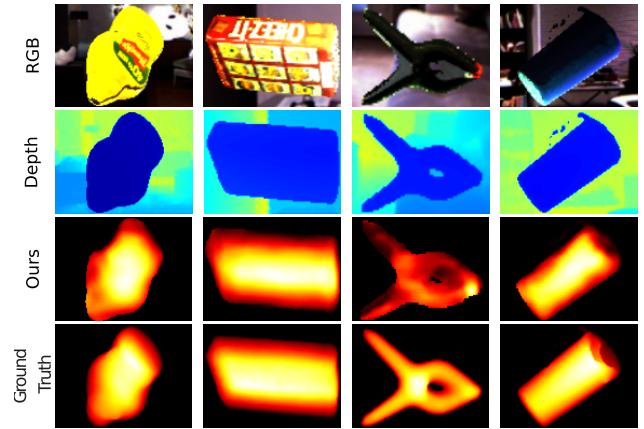


Figure 6: Examples of training data with prediction of the synthetic YCB dataset. Objects are hard to recognise because of domain randomisation and subsampling. Thickness is predicted by one of the fully trained networks of which the performances are reported below.

[36] suggests that randomisation of certain attributes leads to the robustification of the learning with respect to that characteristic. Hence, we render objects with random number of lights, intensity, colour and positions. This domain-randomisation approach aims to guide the network to ignore environmental features and focus on shape cues.

Our rendering pipeline renders depth and RGB at a resolution of 640×480 with objects at a random distance from the camera. We then crop the image using the bounding box of the object and resize the crop using bilinear sampling, simulating the object detection process. To add more realistic background we placed the rendered object in front of RGB and depth frames randomly picked from the NYU dataset [26]. The resulting dataset comprises 2000 images per modality for 86 of the objects in the YCB dataset. Figure 6 shows a sample of the training dataset along with network prediction and ground truth cross-section. Cross-

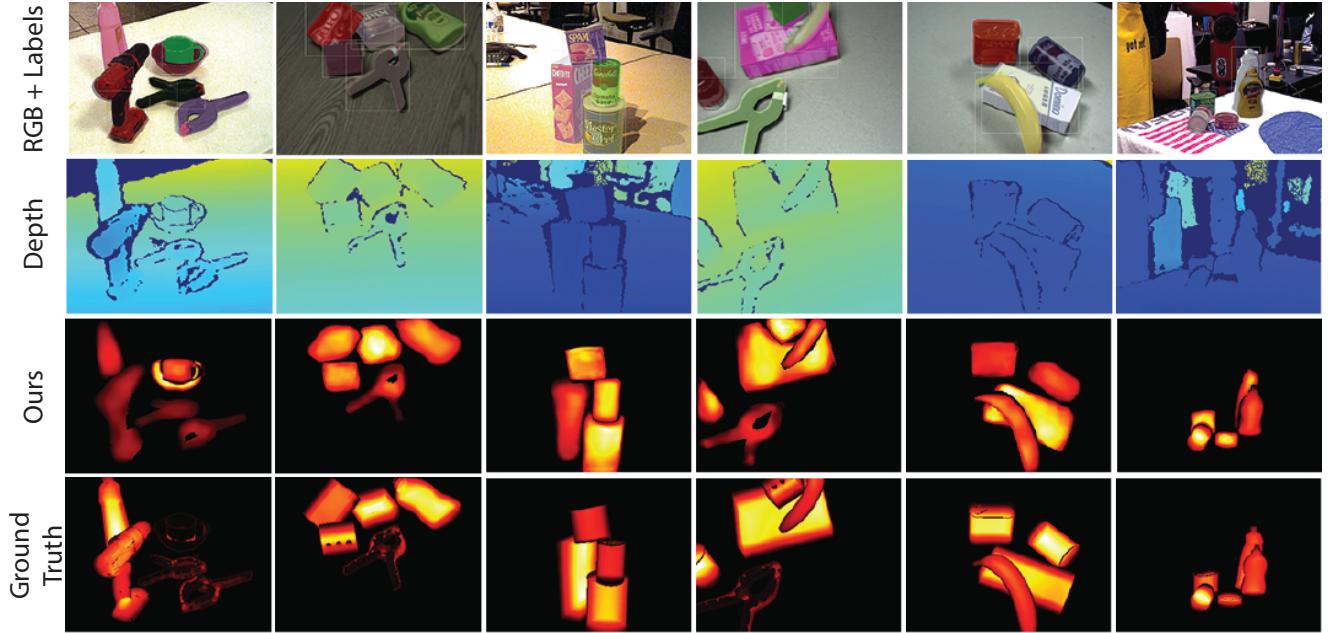


Figure 7: Prediction on the YCB Video dataset using ground truth bounding boxes and segmentation.

sectional thickness is rendered with a custom shader in Blender¹. By design the shader returns only the visible surface thickness. Subsequent surfaces are ignored. This choice is inspired by our focus on multi-view fusion. Our approach allows for the incremental refinement of an object by fusing predicted thickness over multiple views. By not predicting the thickness of unobserved surfaces, we avoid integrating wrong information from hallucinated structures.

To bridge the gap between real and synthetic data, we fine tune the network on the YCB Video dataset presented in [43]. The dataset is composed of 90 videos of table top scenes captured with an Asus Xtion Pro Live. Every RGB and depth image is accompanied by semantic labels, bounding boxes and poses of the objects relative to the cameras. We take advantage of such information to replicate the scene in Blender and render the thickness frame. We then use bounding boxes and labels to crop patches of single objects from depth and thickness and to create the corresponding silhouettes. In this way we render 100 thickness images for each of 80 of the videos.

5. Results

To analyse the effectiveness of the approach, we trained X-Section and design three experiments. In 2D we compare against the validation set. Since our method predicts unseen information from RGB-D frames, it can be seen as a shape completion problem. Hence, we benchmark our pipeline against Voxlets [12]. Finally, we fuse multiple predictions

and show the difference with respect to a voxelised representation of the scene.

The ResNet backbone is pre-trained on ImageNet and the whole network is trained for 40 epochs, with learning rate of $1e - 5$ and batch of 50, 128x92 images. We reserve ten percent of the dataset as validation set. The model is then fine-tuned on data from YCB Video leaving out 12 sequence for validation. We found 10 epochs to be sufficient to achieve satisfactory results.

5.1. 2D Evaluation

To the best of our knowledge there is no related method that has been proposed to predict the cross-sectional thickness of objects. Thus, we adopt the mean thickness over all pixels of the objects in the training set as reference. We test two variants of X-Section, one with ResNet50 and with ResNet101 backbone. Both networks are trained on the same amount of data for the same number of epochs. We define t_p and \hat{t}_p as the ground truth and predicted thickness, respectively. Over N pixels we compute the metrics:

$$\text{Abs. Relative Difference} = \frac{1}{N} \sum_p \frac{|t_p - \hat{t}_p|}{t_p}, \quad (2)$$

$$\text{Square Relative Difference} = \frac{1}{N} \sum_p \frac{\|t_p - \hat{t}_p\|^2}{t_p},$$

$$\text{Log Root Mean Square} = \sqrt{\frac{1}{N} \sum_p \|\log t_p - \log \hat{t}_p\|^2}.$$

¹<https://www.blender.org/>

	Baseline	Ours			
		ResNet 101		ResNet 50	
		DS	RGB-D	DS	RGB-D
absolute relative difference	96.044	3.819	4.301	3.896	4.047
sqr relative difference	4.074	0.047	0.056	0.045	0.059
RMSE (linear)	0.026	0.015	0.015	0.013	0.014
RMSE (log)	1.545	0.700	0.693	0.671	0.689

Table 1: 2D evaluation results on the YCB-video dataset. Thickness is measured in meters. We test different inputs, Depth with Silhouette (DS) and RGB with Depth (RGB-D). The baseline is the mean thickness over the training dataset.

The results are gathered in Table 1. As expected the network performs better than mean on all tests. It can be noticed that the performance gap between two different versions of X-Section is not significant. This hints that breaking down the scene in smaller components simplifies the task, requiring a smaller network. A more thorough investigation is required to draw conclusions about this and left as future work. Large values of the *absolute relative difference* of the baseline are the result the view-centred formulation of the task that makes the data dependent on the incidence angle of the observation ray. As a consequence the value of thickness tends to zero at the border of objects where rays are tangent to the surface. The fact that X-Section produces such low values for this metric suggests that the network has actually learnt to predict the shape coherently.

5.2. RGB-D Vs. Depth and Silhouette

To isolate where most of the information is stored, we have trained a network with RGB and depth as input and one with a depth image and a silhouette. As shown in Table 4 and Table 2 the use of RGB and depth causes a drop in performance. When a mask is passed in input, the network takes the mask into account when making predictions and this guides the learning to better exploit the information stored in the pixels picturing the object.

Although in principle the RGB data should hold important information for shape reconstruction, this type of input is the one that suffer from domain adaptation the most. It is also to be considered that depth retains direct information of the shape and it might cause the network to ignore cues in colour data. This analysis leans in favour of the use of 2.5D sketches for shape recovering. However, a stronger conclusion on the best input for this type of algorithms requires a more thorough and precise analysis that is out of the scope of this work.

5.3. Comparison with Voxlets

Our focus is to retrieve geometric information from an incomplete measurement of the environment. This makes this work closely related to 3D shape completion, such as [11] or [12]. The voxel resolution of the former approach

is 5cm making it hard to directly test it in table top scenarios. On the contrary, Voxlets [12] is showcased in table top scenes and provides trained models and data.

We run our pipeline on the dataset released with [12] and we pick eight scenes with highest detection rate. As ground truth we use the voxel grids provided. Most of the instances are completely new to the network and their shape non trivial. Examples of objects of the dataset are boxes, shoes, a teapot and a cast head. We think this difficult scenario thoroughly tests the generalisation capabilities of the network. We run our pipeline on a single frame and compare our single-view reconstruction with the 3D completion approach in Voxlets. Figure 10 shows the scene reconstructed with our method, our implementation of a depth only fusion algorithm, the output of Voxlets and reference complete volume.

After fusing the predictions in a TSDF volume using the algorithm described in Section 3.4, we recover occupancy values by binarising the obtained TSDF values in the 3D grid. We classify voxels as occupied if the TSDF values are less than the truncation value τ and free otherwise. Calling \mathcal{V}_g the ground truth volume and \mathcal{V}_x the volume reconstructed with X-Section predictions, Intersection Over Union, precision and recall can be computed as

$$IoU = \frac{\mathcal{V}_g \cap \mathcal{V}_x}{\mathcal{V}_g \cup \mathcal{V}_x}, \quad P = \frac{p_t}{p_t + n_t}, \quad R = \frac{p_t}{p_t + n_f}.$$

Where p_t is the number of true positive predictions (so a voxel correctly predicted as belonging to the object volume), n_t denotes the number of true negatives and n_f the number of false negatives.

Table 4 shows X-Section falling short of few percentage points with respect to the baseline. There are several reasons behind the accuracy of our approach on this data. A crucial factor is that the objects used for this benchmark do not compare to the ones in the dataset, hence the network is seeing not only a novel view, but also a novel model and novel class for all inputs. Moreover, our approach does not complete the scene where there are no depth readings. This yields to incomplete reconstruction when objects are

	Baselines		Ours (X-Section)			
	Voxlets	DF	ResNet 101 base		ResNet 50 base	
			DS	RGB-D	DS	RGB-D
IoU	0.713	0.327	0.761	0.620	0.759	0.651
Precision	0.893	0.887	0.894	0.875	0.837	0.882
Recall	0.779	0.341	0.836	0.680	0.890	0.713

Table 2: Results of the comparison against Voxlets [12] for sequences with all objects detected. As baseline we adopt Voxlets and our implementation of a depth only fusion algorithm via TSDF averaging (DF).

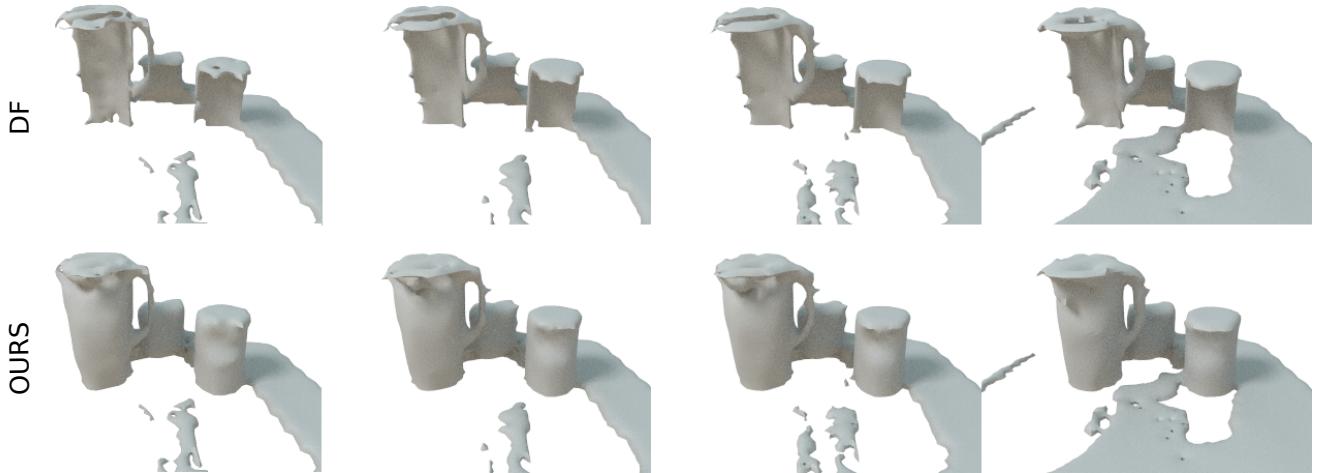


Figure 8: Proposed enhanced fusion in a YCB Video sequence. Top row, fusion of depth frame with TSDF averaging (DF). Bottom row, the proposed augmented fusion. We chose spatially distant frames. From left to right, fusion of frame 0, 60, 120 and 270.

occluded. On the other hand, Voxlets tries to fill the gaps, scoring better in the chosen metrics.

To investigate the impact of a faulty object detector, we ran the pipeline on a sequence where all objects are successfully segmented. As Table 2 shows, in this case the accuracy of the prediction is beyond what Voxlets achieves; showing impressive generalisation capabilities. The use of an object detection stage results in a trade off in terms of generality. Isolating the single objects is portable across different scenarios and environments without requiring any retraining or fine tuning. Voxlets, however, needs to be trained on every different scene type.

5.4. Multi-Frame Fusion Evaluation

The main application of the X-Section pipeline is the integration of thickness prediction in a multi-frame fusion system. The YCB Video dataset [43] provides relative poses of object with respect to the camera. We use this information to compose the scene and produce a solid voxelisation to be used as ground truth approximation. Figure 7 shows the result of our pipeline on sample frames of the validation

dataset. Using the algorithm in Section 3.4 we fuse the predictions for the first 50 frames of each of the 12 validation sequences.

Figure 9 reports the metrics computed per each frame fused from sequence 0052 and 0048 of the dataset. In this two scenes we report IoU and recall almost twice as high as the ones obtained by fusing only depth frames. This is a consequence of reconstructing explicitly the volume and not only the surface as traditional TSDF fusion algorithms do. However, it is also important that we recover accurately the shape of the object. This is reflected by the precision metric. On this specific case 90% of the voxels recovered are true positives, matching the performance of depth only fusion that uses only sensor readings.

Table 3 reports the average value for all the metrics for every validation sequence. IoU and Recall rates are always in favour of the suggested pipeline. On some sequences, our approach falls slightly short in terms of precision. Since the proposed method predicts unseen surfaces in difficult scenes the network predicts a small percentage of false positives. This drawback could be mitigated by predicting a

	0048		0049		0050		0051		0052		0053	
	DF	Ours	DF	Ours	DF	Ours	DF	Ours	DF	Ours	DF	Ours
IoU	0.299	0.535	0.346	0.513	0.233	0.392	0.355	0.735	0.264	0.693	0.252	0.395
Precision	0.787	0.841	0.659	0.872	0.804	0.894	0.901	0.911	0.881	0.484	0.535	
Recall	0.326	0.596	0.393	0.698	0.241	0.433	0.371	0.780	0.271	0.764	0.345	0.600

Table 3: Evaluation of multi-frame fusion averaged over the first 50 frames of the YCB Video dataset [43]. We compare our modified TSDF fusion of Section 3.4 and a depth only fusion algorithm, labelled DF.

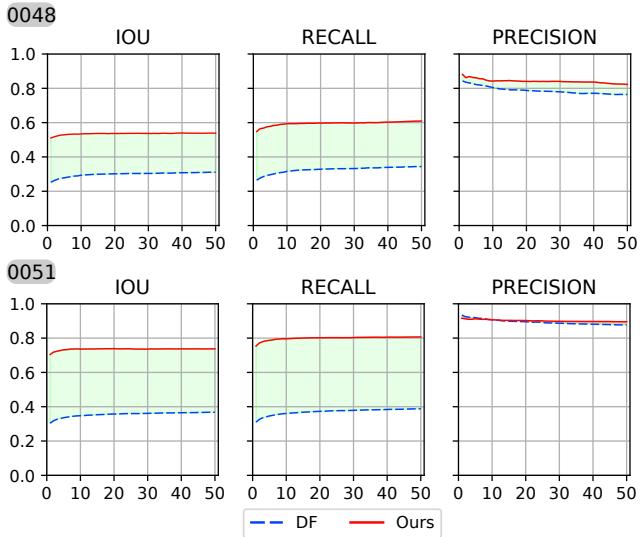


Figure 9: Evaluation of multi-frame fusion for two of the sequences of the YCB Video dataset [43] reserved for validation. Top sequence 0048, bottom 0051. The red solid line represent the result of our method, the blue dashed line shows the performance of a depth only fusion algorithm.

per pixel uncertainty and use it for probabilistic mapping. Investigations in this direction are reserved for future work.

Figure 8 reports the result of multiple-view fusions of another validation sequence. The reconstructed scene is shown from the back of the observed surfaces. The frames are relatively spatially distant for a table top scene. The bottom row shows the result of the thickness fusion algorithm described in Section 3.4. The results shows consistent predictions and over time the reconstruction quality improves. Whenever there is no thickness information (such has the table surface) only depth is fused (i.e. with traditional TSDF).

6. Conclusions And Future Work

In this work we have presented the novel task of predicting the cross-sectional thickness of objects in a scene. We introduced a model for solving this task that involves decomposing a scene into individual objects, predicting the thickness and then recomposing the scene. Our experiments

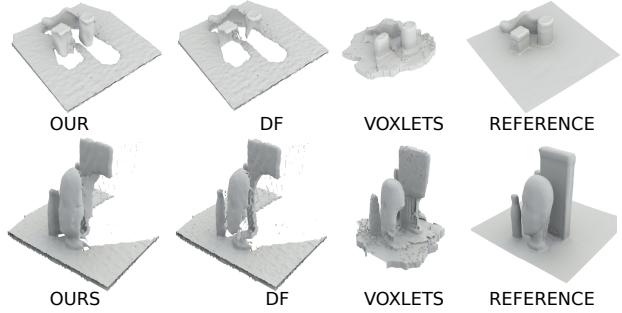


Figure 10: Reconstruction results and comparison with Voxlets. Each row shows two different reconstructed scene. From left to right: results of our fusion algorithm using predicted thickness, results of a depth only fusion via TSDF averaging (DF), the output of Voxlets and the reference model.

	Voxlets	DF	Ours (ResNet 101 - DS)
IoU	0.622	0.234	0.440
Precision	0.811	0.695	0.703
Recall	0.735	0.261	0.536

Table 4: 3D evaluation of our approach on the Voxlets dataset for eight sequences on which Mask R-CNN has missing detections. We show comparisons against Voxlets and depth only fusion via TSDF averaging (DF).

show that we can train our model and recover the 3D shape of the object with a simple extension to traditional fusion algorithms. To overcome the difficulties of domain adaptation we fine tuned on real world images. This proved to be central for test time performances.

We demonstrated the convenience and compactness of predicting the cross-sectional thickness of objects and it's usefulness in reconstruction scenarios. Moreover, predicting one layer only has the advantage of limiting the estimation to observed surfaces, avoiding inaccuracy caused by the network hallucinating non observable parts of the scene. On the other hand this might yield incomplete models. There are different ways to approach this issue and we aim to investigate some in future works.

References

- [1] Bruce G Baumgart. A polyhedron representation for computer vision. In *Proceedings of the May 19-22, 1975, national computer conference and exposition*, pages 589–596. ACM, 1975. 2
- [2] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016. 3
- [3] Berk Calli, Aaron Walsman, Arjung Singh, Siddhartha Srinivasa, Peter Abbeel, and Aaron M. Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics Automation Magazine*, 22(3):36–52, Sept 2015. 2, 4
- [4] Hao Chen, Xiaojuan Qi, Lequan Yu, Qi Dou, Jing Qin, and Pheng-Ann Heng. Dcan: Deep contour-aware networks for object instance segmentation from histology images. *Medical image analysis*, 36:135–146, 2017. 3
- [5] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. Masklab: Instance segmentation by refining object detection with semantic and direction features. *arXiv preprint arXiv:1712.04837*, 2017. 3
- [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 *European conference on computer vision*, pages 628–644. Springer, 2016. 3
- [7] Ronald Clark, Michael Bloesch, Jan Czarnowski, Stefan Leutenegger, and Andrew J Davison. Learning to solve non-linear least squares for monocular stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299, 2018. 2
- [8] Antonio Criminisi, Ian Reid, and Andrew Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000. 3
- [9] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996. 2, 4
- [10] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(4):76a, 2017. 2
- [11] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *CVPR*, volume 1, page 2, 2018. 2, 6
- [12] Michael Firman, Oisin Mac Aodha, Simon Julier, and Gabriel J Brostow. Structured prediction of unobserved voxels from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5431–5440, 2016. 2, 5, 6, 7
- [13] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *3D Vision (3DV), 2017 International Conference on*, pages 402–411. IEEE, 2017. 3
- [14] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016. 3
- [15] JunYoung Gwak, Christopher B Choy, Manmohan Chandraker, Animesh Garg, and Silvio Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In *3D Vision (3DV), 2017 International Conference on*, pages 263–272. IEEE, 2017. 3
- [16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017. 3
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3
- [18] Derek Hoiem, Alexei A Efros, and Martial Hebert. Automatic photo pop-up. In *ACM transactions on graphics (TOG)*, volume 24, pages 577–584. ACM, 2005. 3
- [19] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 225–232. ACM Press/Addison-Wesley Publishing Co., 1997. 3
- [20] Andrey Kurenkov, Jingwei Ji, Animesh Garg, Viraj Mehta, JunYoung Gwak, Christopher Choy, and Silvio Savarese. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. *arXiv preprint arXiv:1708.04672*, 2017. 3
- [21] Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. In *British Machine Vision Conference (BMVC)*, 2018. 4
- [22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, 2014. 3
- [23] Charles Loop, Qin Cai, Sergio Orts-Escolano, and Philip A. Chou. A closed-form bayesian fusion equation using occupancy probabilities. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 380–388, Oct 2016. 1
- [24] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level slam. In *2018 International Conference on 3D Vision (3DV)*, pages 32–41. IEEE, 2018. 2
- [25] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML’10*, pages 807–814, 2010. 3
- [26] Pushmeet Kohli, Nathan Silberman, Derek Hoiem, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 4

- [27] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, Oct 2011. [2](#), [4](#)
- [28] Mukta Prasad and Andrew Fitzgibbon. Single view reconstruction of curved surfaces. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1345–1354. IEEE, 2006. [3](#)
- [29] Mukta Prasad, Andrew W Fitzgibbon, and Andrew Zisserman. Fast and controllable 3d modelling from silhouettes. In *Eurographics (Short Presentations)*, pages 9–12, 2005. [2](#)
- [30] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. Octnetfusion: Learning depth fusion from data. In *3D Vision (3DV), 2017 International Conference on*, pages 57–66. IEEE, 2017. [2](#)
- [31] Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision*, pages 236–250. Springer, 2016. [3](#)
- [32] Daeyun Shin, Charless Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [2](#)
- [33] Daeyun Shin, Zhile Ren, Erik B Sudderth, and Charless C Fowlkes. Multi-layer depth and epipolar feature transformers for 3d scene reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. [3](#)
- [34] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [2](#), [4](#)
- [35] Niko Sünderhauf, Trung T Pham, Yasir Latif, Michael Milford, and Ian Reid. Meaningful maps with object-oriented semantic mapping. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 5079–5085. IEEE, 2017. [2](#)
- [36] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *The European Conference on Computer Vision (ECCV)*, September 2018. [4](#)
- [37] Alexandru Telea. An image inpainting technique based on the fast marching method. *J. Graphics, GPU, Game Tools*, 9:23–34, 2004. [3](#)
- [38] Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. *Computer Vision and Pattern Recognition (CVPR)*, 2018. [3](#)
- [39] Emanuele Vespa, Nikolay Nikolov, Marius Grimm, Luigi Nardi, Paul H. J. Kelly, and Stefan Leutenegger. Efficient octree-based volumetric slam supporting signed-distance and occupancy mapping. *IEEE Robotics and Automation Letters*, 3(2):1144–1151, April 2018. [2](#)
- [40] Chamara Saroj Weerasekera, Yasir Latif, Ravi Garg, and Ian Reid. Dense monocular reconstruction using surface normals. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2524–2531. IEEE, 2017. [2](#)
- [41] Thomas Whelan, Michael Kaess, Maurice Fallon, Hordur Johannsson, John Leonard, and John McDonald. Kintinuous: Spatially extended kinectfusion. 2012. [2](#)
- [42] 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 *Advances in Neural Information Processing Systems*, pages 82–90, 2016. [3](#)
- [43] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017. [2](#), [5](#), [7](#), [8](#)
- [44] Bo Yang, Hongkai Wen, Sen Wang, Ronald Clark, Andrew Markham, and Niki Trigoni. 3d object reconstruction from a single depth view with adversarial learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 679–688, 2017. [2](#)
- [45] Li Zhang, Guillaume Dugas-Phocion, Jean-Sebastien Samson, and Steven M Seitz. Single-view modelling of free-form scenes. *The Journal of Visualization and Computer Animation*, 13(4):225–235, 2002. [3](#)
- [46] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Joshua B Tenenbaum, William T Freeman, and Jiajun Wu. Learning to Reconstruct Shapes from Unseen Classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. [3](#)