

Predicting Structured Geometry From Single And Multiple Views

Anonymous ECCV submission

Paper ID 1485

Abstract. We develop a structured prediction approach to reconstructing 3D polygonal models from single and multiple images of a scene. Building on recent advances in single view reconstruction we adopt the indoor Manhattan hypothesis class — one of the most complicated output spaces (in terms of internal constraints) yet considered within a structured prediction framework. Our approach can learn in both the single- and multiple-view contexts. We show that the chosen hypothesis class permits optimizing a variety of high-level loss functions, such as the relative depth error. Our results out-perform the state-of-the-art, including an improvement of more than 50% on one metric.

1 Introduction

Reconstruction 3D models from images is a central problem in computer vision. There is a deep literature concerning reconstruction from multiple views of a scene, with particular focus on the geometry of multiple cameras. More recently the recovery of 3D information from a single image has received attention [?,?]. In this context the geometric constraints leveraged by multiple-view reconstructors are unavailable, so inference must rely on photometric cues. To capture the complex, high-dimensional patterns involved in this challenging inference task, practitioners have leveraged a range of machine learning techniques.

Within single-view reconstruction, few authors have cast learning as a single optimisation with respect to a clearly defined loss function [?,?,?,?], while most approaches to multiple-view reconstruction do not consider learning from training data at all. In contrast, this paper casts reconstruction fundamentally as a learning problem, with the goal being to learn a prediction function f mapping observed features to 3D reconstructions.

Building on recent advances in single-view reconstruction we adopt as our hypothesis class the set of indoor Manhattan models [?,?,?], under which scenes are approximated by a floor and ceiling plane together with a sequence of vertical walls. This representation brings a variety of attractive features such as a simple parametrisation, efficient and exact inference, decomposability of loss functions, and a balance between expressiveness and robustness.

For learning we employ the tools of structured prediction, in particular the structural SVM [?]. The use of these tools is part of a long trend towards statistically rigorous, well-understood convex optimization techniques in computer vision. Recent successfully applications include detection [?], segmentation

[?], and scene classification [?]. In the domain of reconstruction, structured prediction ideas have been applied to several simple models classes such as stereo disparities [?] and cuboids [?].

The application of structured prediction to the indoor Manhattan class of models constitutes one of the most complex output spaces yet considered within this framework. The indoor Manhattan model enforces hard geometric constraints that lack simple expressions in terms of image coordinates. These constraints are context-dependent, being tied to quantities such as camera rotation and the location of vanishing points. We have learnt several valuable lessons of general relevance from this complex prediction task, to which we dedicate the final section of this paper.

The contributions of this paper are thus (i) a unified learning framework for single- and multiple-view reconstruction, utilising the indoor Manhattan model and the tools of structured prediction; (ii) the reduction of two image-level loss functions to a form amenable to efficient optimization; (iii) an efficient separation procedure for identify the “most-violated constraint” during learning, (iv) an empirical demonstration of structured prediction in perhaps the most complex output space yet considered within this framework; and (v) a series of practical observations concerning the application of structured prediction techniques.

In the remainder of this paper we present background material (Section 2), followed by the indoor Manhattan model itself (Section 3), and our learning framework (Section 4). We then present results for multiple-view reconstruction (Section 5) followed by single-view reconstruction (Section 6), then we close with practical lessons learnt (Section 7) and concluding remarks (Section 8).

2 Background

This paper touches on several major research areas (multiple-view reconstruction, single-view reconstruction, and structured prediction), so here we present only key contributions and those results of particular relevance to our own work.

Multiple view reconstruction has a long history in the literature, beginning at least as early as the seminal work of Marr *et al.* [?]. Low-level approaches estimate the depth of each pixel by solving for stereo disparity (for a survey see [?]), while high-level approaches attempt to recover polyhedral models of various kinds [?,?]. The present work follows the spirit of the latter approach.

Coughlan and Yuille [?] first introduced the Manhattan world assumption, in which reconstructed surfaces are restricted to three mutually-orthogonal orientations. Furukawa *et al.* [?] applied this idea within a multiple view stereo context. Lee *et al.* [?] proposed the indoor Manhattan assumption, which places further restrictions on reconstructed models.

Single-view reconstruction from line drawings also has a long history in the literature ([?], for example). Reconstructing single real-world images was re-introduced to the community by Hoiem *et al.* [?], who employed decision trees to classify image segments into orientation classes in a multiple segmentation approach. Saxena *et al.* [?] employed energy minimisation to recover pixel-wise

depths. Barinova *et al.* [?] reconstructed piece-wise planar outdoor scenes using an EM algorithm. Hedau *et al.* [?] modelled indoor scenes as the interior of a cuboid, and cast learning within a structured prediction framework. We also use structured prediction, though our hypothesis class is far more expressive.

Lee *et al.* [?] were the first to reconstruct indoor Manhattan scenes. They devised a combinatorial search over line segments using a branch-and-bound algorithm. Flint *et al.* [?] refined this to an exact dynamic programming solution, and in later work [?] extended the indoor Manhattan assumption to the multiple view domain. The authors of [?] did describe a learning algorithm based on bootstrapping, though its statistical consistency was given little attention. The present work relies extensively on this thread of research, though in contrast to past work our focus is entirely on learning in a statistically rigorous framework. We provide comparisons with these approaches.

Extending binary and multi-label classification to general output spaces is a major research programme within the machine learning community; an excellent introduction is given by Bakir *et al.* [?]. Our approach employs the structured support vector machine, first proposed by Tsochantaridis *et al.* [?] and improved upon by Joachims *et al.* [?].

3 Model

In this section we describe three components of the model that we are trying to learn (and, at test time, infer): a hypothesis class, a feature space, and a loss function. In our setup these are, respectively, the class of indoor Manhattan models, a log-linear Bayesian likelihood, and either the relative depth error or a labelling error (we describe both).

3.1 Hypothesis Class

This paper is concerned with the hypothesis class consisting of indoor Manhattan reconstructions, which are 3D polygonal models characterized by infinite floor and ceiling planes with vertical walls extending between them (as originally proposed by Lee *et al.* [?]). Indoor Manhattan environments are a sub-class of general Manhattan environments; examples are shown in figure Figure 1. This is an attractive hypothesis class because

1. it captures many regularities within man-made environments;
2. the geometric primitives (floor/wall/ceiling) are immediately useful for semantic-level scene understanding;
3. it is expressive enough to represent approximately or exactly a surprisingly wide variety of environments;
4. there is a simple and convenient parametrisation;
5. an efficient inference algorithm exists [?].

An indoor Manhattan model is much like an architectural floor-plan, and can be specified as a set of 2D line segments representing walls together with

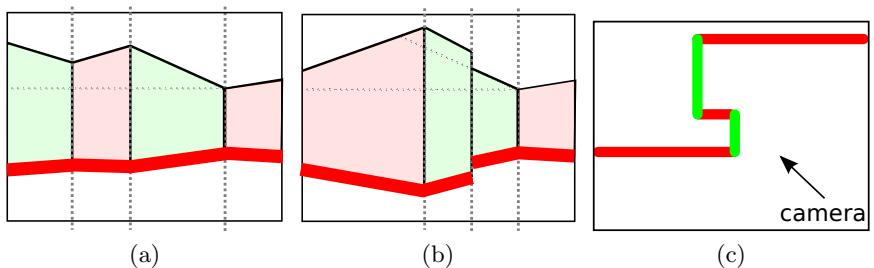


Fig. 1: (a-b) Examples of indoor Manhattan environments. The red line illustrates the seam representation. (c) An indoor Manhattan environment viewed from above.

the position of the floor and ceiling plane (Figure 1). In this paper we adopt an image-domain parametrisation due to its convenience for inference and learning. Following Flint *et al.* [?] we represent hypotheses as a path running from the left edge to the right edge of the image, which we will refer to as a *seam*. A seam S consists of a sequence of pairs of scalars,

$$S = \{s_i, o_i\}_{i=1}^W \quad (1)$$

where s_i is the y -coordinate at which the path intersects image column i , o_i is the orientation of the wall in that column, and W is the image width. Remarkably, this simple parametrisation specifies a unique metric 3D model up to scale [?].

While we do not here have space for a full discussion of the geometry of indoor Manhattan models, there are two properties of the seam representation that will be relevant in the following sections. Firstly, images of indoor Manhattan environments can be rectified so that vertical lines in the world project to vertical lines in the image [?]. Secondly, the mapping from seam to 3D reconstruction decomposes in the following manner. We said earlier that a seam S specifies a unique 3D reconstruction. Let $a(x, y; S)$ be the orientation of the 3D surface projecting to pixel (x, y) . Flint *et al.* [?] showed that a is functionally dependant *only* on the pair (s_x, o_x) for column x . That is, we may write

$$a(x, y; S) = \tilde{a}(x, y; s_x, o_x) . \quad (2)$$

Similarly, letting $d(x, y; S)$ be the distance from the camera to the surface projecting to (x, y) we may write

$$d(x, y; S) = \tilde{d}(x, y; s_x, o_x) . \quad (3)$$

3.2 Feature Space

We adopt the probabilistic model formulated by Flint *et al.* [?], which relates indoor Manhattan reconstructions to single-view image features, multiple-view

photo-consistency terms, and a reconstructed point cloud. From our perspective these are simply features and any subset may be omitted if inappropriate for a given application. Under this model the prior on reconstructions is

$$\log P(S) = -\mathbf{n} \cdot \boldsymbol{\lambda} + O(1) \quad (4)$$

where \mathbf{n} is a vector containing the number of walls in S of various categories (concave, convex, occluding) and $\boldsymbol{\lambda}$ is a hyper-parameter. We may interpret (4) as, roughly, that reconstructions with more walls are less likely.

Flint *et al.* [?] showed that for a variety of features $\boldsymbol{\theta}$ (including single-view and multiple-view) there are reasonable choices of likelihood that can be written

$$\log P(\boldsymbol{\theta} | S) = \sum_{x=1}^W \pi_{\boldsymbol{\theta}}(x, s_x) \quad (5)$$

where $\pi_{\boldsymbol{\theta}}$ is a real matrix computed deterministically from $\boldsymbol{\theta}$ and $\pi_{\boldsymbol{\theta}}(i, j)$ is the element at row i , column j . It is easy to show that for each sensor model described in [?] the log-likelihood is linear in the hyper-parameters $\boldsymbol{\kappa}$, *i.e.*

$$\log P(\boldsymbol{\theta} | S) = \sum_{x=1}^W \boldsymbol{\kappa}_{\boldsymbol{\theta}} \cdot \nu_{\boldsymbol{\theta}}(x, s_x) . \quad (6)$$

Assuming conditional independence between features, the posterior is

$$P(S | \Theta) \propto P(S) \prod_{i=1}^n P(\boldsymbol{\theta}_i | S) \quad (7)$$

$$\log P(S | \Theta) = -\mathbf{n} \cdot \boldsymbol{\lambda} + \sum_{i=1}^n \sum_{x=1}^W \boldsymbol{\kappa}_i \cdot \nu_i(x, s_x) + O(1) . \quad (8)$$

So far, this model closely follows that described in [?]; our contribution is to place this model into a statistically rigorous learning framework. To do this we need to rewrite the above in terms of a joint feature function $\Psi(\Theta, S)$ and a parameter vector \mathbf{w} . The decomposability of indoor Manhattan models into payoff matrices permits precisely such a formulation. Defining

$$\Psi(\Theta, S) = \begin{bmatrix} -\mathbf{n} \\ \sum_{x=1}^W \nu_1(x, s_x) \\ \vdots \\ \sum_{x=1}^W \nu_n(x, s_x) \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\kappa}_1 \\ \vdots \\ \boldsymbol{\kappa}_n \end{bmatrix} \quad (9)$$

we see that (8) can be written as

$$\log P(S | \Theta) = \langle \mathbf{w} , \Psi(\Theta, S) \rangle + O(1) \quad (10)$$

where we have adopted Hilbert space notation with $\langle \cdot, \cdot \rangle$ denoting an inner product. Since \mathbf{w} contains all free parameters in the model, the goal of learning will be to optimise \mathbf{w} with respect to a training set $\{(\Theta_i, S_i)\}$.

Feature	Dimensionality	Multi-view?	Single-view?	Reference
Stereo photo-consistency	4	yes	no	Flint <i>et al.</i> [?]
Point cloud	2	yes	no	Flint <i>et al.</i> [?]
Line sweeps	1	yes	yes	Lee <i>et al.</i> [?]
RGB+HSV	6	no	yes	
Gabor responses ¹	12	no	yes	

Fig. 2: The composition of our single- and multiple-view feature space. We omit color and Gabor features from the multiple-view feature space for training efficiency.

Features The precise make-up of the feature space depends on the available sensor modalities. We define separate feature spaces for the single- and multiple-view contexts; these are summarized in Figure 2.

3.3 Loss Functions

Next we define a loss function $\Delta(S, \hat{S})$ measuring the cost of predicting some reconstruction \hat{S} when in fact the true reconstruction is S . In the context of learning one often faces a trade-off between choosing a loss that leads to tractable optimization, and choosing a loss that measures the quantity that one “really” cares about. For example, Hoiem *et al.* [?] learn a per-segment orientation classifier, then pass this as input to a separate 3D reconstruction system [?]. However, what one “really” cares about is some loss defined on the output of the entire system rather than the output of individual components, since some segment-level mistakes are insignificant to the overall reconstruction quality, while others are catastrophic. This is not a criticism of the authors’ choice, but an illustration of the trade-off faced when choosing a loss. In this paper we show how to learn efficiently with respect to a loss defined on the final reconstruction.

The relative depth error has been the gold standard within the reconstruction community for more than a decade [?], and measures the average deviation between reconstructed and ground truth depths. In our notation,

$$\Delta_{\text{depth}}(S, \hat{S}) = \frac{1}{N} \sum_{\mathbf{p}} \frac{|d(\mathbf{p}; \hat{S}) - d(\mathbf{p}; S)|}{d(\mathbf{p}; S)} , \quad (11)$$

where N is the number of pixels. Another reasonable choice is the labelling error, used widely within the semantic segmentation literature,

$$\Delta_{\text{labelling}}(\hat{S}, S) = \frac{1}{N} \sum_{\mathbf{p}} [a(\mathbf{p}; \hat{S}) \neq a(\mathbf{p}; S)] , \quad (12)$$

where $[p]$ is 1 if p is true and 0 otherwise. An attractive characteristic of the indoor Manhattan class is that *both of these losses can be optimized exactly*. The

¹ 4 orientations, 3 scales

algorithmic details are left to Section 4; the key result we establish here is that Δ_{depth} and $\Delta_{\text{labelling}}$ can be written in a form resembling the payoff formulation (5) for the feature likelihoods.

First we invoke the independence established in (3):

$$\Delta_{\text{depth}}(\hat{S}, S) = \frac{1}{N} \sum_{x=1}^W \sum_{y=1}^H \frac{|\tilde{d}(x, y; \hat{s}_x) - d(x, y; S)|}{d(x, y; S)} . \quad (13)$$

Defining a real matrix δ_S ,

$$\delta_S(x, j) = \sum_{y=1}^H \frac{|\tilde{d}(x, y; j) - d(x, y; S)|}{d(x, y; S)} , \quad (14)$$

we see that we can write Δ_{depth} in the form

$$\Delta_{\text{depth}}(\hat{S}, S) = \frac{1}{N} \sum_{x=1}^W \delta_S(x, \hat{s}_x) . \quad (15)$$

There is a similar form for $\Delta_{\text{labelling}}$, which we omit here due to space constraints.

Choosing a Loss Function Neither of the above losses is unequivocally the “correct” loss; the choice will depend on the application. One might expect a strong correlation between the losses, and indeed one can show analytically that

$$\Delta_{\text{depth}}(\hat{S}, S) = 0 \iff \Delta_{\text{labelling}}(\hat{S}, S) = 0 . \quad (16)$$

However, in our experiments we found only a weak correlation between these losses away from the origin. For example, the scatter plot shown in figure Figure 5 shows a significant number of outliers that score very well on Δ_{depth} but poorly on $\Delta_{\text{labelling}}$, and vice versa.

4 Learning

We turn now to the problem of learning within the model described above. Our learning task is to identify a prediction function f mapping observed features Θ to reconstructions S . We seek the loss minimizer

$$f^* = \operatorname{argmin}_f \mathbb{E}[\Delta(f(\Theta), S)] , \quad (17)$$

which we approximate in the framework of empirical risk minimization as

$$f^* = \operatorname{argmin}_f \sum_k \Delta(f(\Theta_k), S_k) , \quad (18)$$

315 where k indexes a training set. To perform this optimization we turn to the tools
 316 of structured prediction [?], and in particular the structured SVM [?]. First we
 317 need to define f . In this paper we consider predictors of the form

$$318 \quad f_{\mathbf{w}}(\Theta) = \operatorname{argmax}_S \langle \mathbf{w}, \Psi(\Theta, S) \rangle. \quad (19)$$

320 Comparing with (10) we see that each predictor of the form (19) is simply
 321 implementing MAP inference under some set of hyper-parameters \mathbf{w} . We now
 322 turn to the optimisation problem itself. Following the standard approach [?] we
 323 cast the learning problem as a constrained optimisation problem,
 324

$$325 \quad \min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{k=1}^n \xi_k \\ 326 \quad \text{s.t. } \forall k, S \neq S_k : \langle \mathbf{w}, \Psi(\Theta_k, S_k) \rangle - \langle \mathbf{w}, \Psi(\Theta_k, S) \rangle \geq \Delta(S, S_k) - \xi_k. \quad (20)$$

328 Tsochantaridis *et al.* [?] described an algorithm for solving this minimisation
 329 that is now used extensively within machine learning and computer vision. To
 330 apply this algorithm here we must solve two inference problems:
 331

- 333 1. *Prediction*. This is the maximization described in (19).
- 334 2. *Separation*. The algorithm described in [?] requires a user-supplied procedure
 335 to find the “most-violated constraint” at each iteration. That is,

$$336 \quad \operatorname{argmax}_S \langle \mathbf{w}, \Psi(\Theta_k, S) \rangle + \Delta(S, S_k). \quad (21)$$

338 Our solutions to both of the above build on the algorithm presented by Flint
 339 *et al.* [?, ?], which is a dynamic programming solution to problems of the form
 340

$$341 \quad \operatorname{argmax}_S \sum_x \pi(x, s_x) - \sum_j \gamma(j; S). \quad (22)$$

344 4.1 Inference (Prediction)

345 We showed in Section 3 that (22) can be written in the form (19), so the
 346 prediction problem is a straightforward application of [?]. This is as expected since,
 347 as we have already remarked, (19) is equivalent to MAP inference on indoor
 348 Manhattan reconstructions, which was precisely the subject of [?].
 349

351 4.2 Loss-Augmented Inference (Separation)

352 It turns out that the separation problem can also be solved using the dynamic
 353 programming algorithm mentioned above, as the following proposition shows.
 354

355 **Proposition 1.** Let (Θ_k, S_k) be a training instance with payoff matrices $\{\pi_i\}$
 356 as defined in (5). Let

$$357 \quad \pi_{\text{aug}} = \delta_{S_k} + \sum_i \pi_i. \quad (23)$$

359 Then the solution to (22) with $\pi = \pi_{\text{aug}}$ is identical to the solution to (21).

360 *Proof.* Direct equivalence of the expressions to be maximized. First substitute
 361 (10) and (15) into (22):
 362

$$\log P(S \mid \Theta) + \sum_{x=1}^W \delta_{S_k}(x, s_x) \quad (24)$$

366 Further substituting (5) and defining γ as in [?] gives
 367
 368

$$\sum_i \sum_{x=1}^W \pi_i(x, s_x) - \sum_j \gamma(j; S) + \sum_{x=1}^W \delta_{S_k}(x, s_x). \quad (25)$$

371 Finally we see that substituting (23) gives
 372
 373

$$\sum_{x=1}^W \pi_{\text{aug}}(x, s_x) - \sum_j \gamma(j; S) \quad (26)$$

377 5 Multiple View Results

379 We evaluated our approach on the data-set proposed in [?], which consists of
 380 18 sequences of six environments averaging 59 seconds in duration. We sampled
 381 key-frames at regular intervals. Each “instance” in our training and hold-out
 382 sets consists of one base frame together with four auxiliary frames.
 383

384 We compared with the bootstrapping approach described in [?]. Our metrics
 385 differ from theirs in two ways. Firstly, they compute relative depth error using
 386 the maximum of the ground truth and estimated depths in the denominator,
 387 whereas we always use the ground truth in the denominator. These metrics are
 388 separated by at most a monotonic transform but the latter is more convenient
 389 to represent in our framework. Secondly, when we compute labelling error we
 390 differentiate vertical and horizontal surfaces only, whereas they also differentiate
 391 the two vertical orientations. The latter approach makes a side-by-side compari-
 392 son difficult because the two vertical orientations are symmetric and their labels
 393 can always be interchanged.

394 The performance of these two algorithms are summarized in Figure 3. Our ap-
 395 proach significantly out-performs the bootstrapping algorithm. Anecdotally we
 396 noticed that much of the improvement resulted from a reduction in catastrophic
 397 failures. This makes sense because we would expect the learning algorithm to
 398 concentrate on reducing those mistakes that result in the largest loss. Some ex-
 399 ample predictions are shown in Figure 7; many more are included in additional
 400 material.

401 6 Single View Results

402 We evaluated our system for single-view reconstruction using the same data-set
 403 described in the previous section. We used the single-view features summarized
 404

Sequence	Depth Error (%)		Labelling Error (%)	
	This Paper ²	Flint <i>et al.</i>	This Paper ³	Flint <i>et al.</i>
ground	4.9	66.6	2.9	10.4
foyer1	6.1	6.6	3.1	3.1
foyer2	4.3	5.4	3.7	4.0
corridor	14.6	52.9	9.5	19.2
mcr	34.0	67.6	15.	16.2
kitchen	16.8	23.6	5.2	6.1
Average	13.4	37.1	6.7	9.8

Fig. 3: Multiple-view reconstruction performance on held-out data, comparing with Flint *et al.* [?]. For unavoidable reasons we use slightly different metrics so our figures differ from those published in [?]. See main text for explanation.

Sequence	Depth Error (%)		Labelling Error (%)	
	This Paper ²	Flint <i>et al.</i>	This Paper ³	Flint <i>et al.</i>
ground	17.3	24.5	7.8	12.2
foyer1	25.1	31.0	15.1	22.2
foyer2	29.1	30.1	15.9	18.6
corridor	31.7	33.6	19.3	24.8
mcr	70.1	45.9	26.7	20.8
kitchen	25.1	26.2	7.7	11.9
Average	33.1	31.9	15.4	18.4

Fig. 4: Single-view reconstruction performance on held-out data, comparing with Flint *et al.* [?].

in Figure 2. We compared our approach to the single-view approach of Flint *et al.* [?], which uses the same dynamic programming algorithm that we rely upon, but uses hand-tuned features.

Performance for each algorithm is summarized in Figure 3. When measured by labelling error, our approach out-performs the hand-tuned weights, but on the depth error metric our approach is inferior. While investigating this result we found that our learning algorithm assigns small weights to all but the line-sweep features, which are the same features used by [?]. This suggests that the hand-tuned weights are in fact close to optimal within this feature space, though one would expect that with additional feature engineering our learning algorithm would be able to leverage further salient information and reduce the error rate.

² This column represents the predictor trained with respect to Δ_{depth} .

³ This column represents the predictor trained with respect to $\Delta_{\text{labelling}}$.

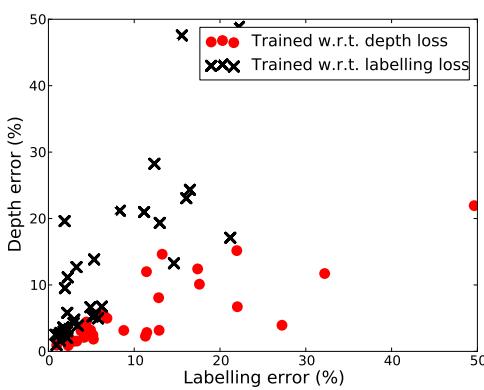


Fig. 5: The effect of the loss function on training. We train two predictors, one with respect to Δ_{depth} and one with respect to $\Delta_{\text{labelling}}$, then evaluate both on all held-out instances. Each data point shows the error obtained by one predictor on held-out instance. The differing distribution of errors shows that the two predictors trade off errors as expected.

7 Discussion

The hypothesis class considered in this paper is amongst the most complex (in terms of internal constraints on the output space) studied within the structured prediction framework. In this section we turn to some practical lessons learnt that may be of value to other practitioners.

Condition in the joint feature space, not the input feature space. A common pre-processing operation for statistical learning is to transform the observed features Θ to zero mean and unit variance. However, for structured

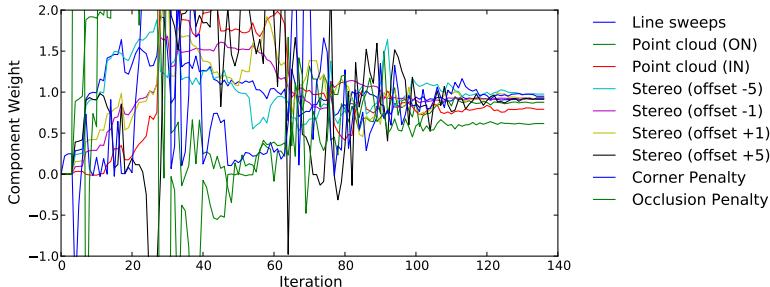


Fig. 6: Evolution of \mathbf{w} during training. Each series shows the value of one component of \mathbf{w} . After an exploration phase the model converges.

prediction tasks it is the joint feature space Ψ that should be conditioned:

$$\Psi' = \frac{\Psi - \mu}{\sigma^2}. \quad (27)$$

Ideally one would sample from the joint feature space to determine the conditioning transformation, but the distribution of inputs and outputs is generally unknown in an empirical risk minimization setting. Instead, we use the training set as a proxy. We compute the empirical mean and variance of $\{\Psi(\Theta_k, S_k)\}_{k=1}^n$ at the outset, then apply the transformation (27) after each feature computation.

Condition the loss terms. For any $\eta > 0$, the minimization problem (20) is equivalent (under the substitution $\mathbf{w}' = \eta \mathbf{w}$, $\xi' = \eta \xi$) to:

$$\begin{aligned} \min_{\mathbf{w}', \xi'} \quad & \frac{1}{2} \|\mathbf{w}'\|^2 + \eta C \sum_{k=1}^n \xi'_k \\ \text{s.t. } \forall k, S \neq S_k : \quad & \langle \mathbf{w}', \Psi(\Theta_k, S_k) \rangle - \langle \mathbf{w}', \Psi(\Theta_k, S) \rangle \geq \eta \Delta(S, S_k) - \xi'_k. \end{aligned} \quad (28)$$

Although any $\eta > 0$ preserves the correctness of the optimization algorithm, we found that choosing $\eta = \text{Var}(\Delta)$ improved numerical stability, since this means the loss terms will have roughly unit variance. Unfortunately, we cannot use the training set to estimate $\text{Var}(\Delta)$ since the loss for the ground truth reconstruction is always zero. Instead we computed $\Delta(\Theta_k, S_j)$ for each $k \neq j$ in the training set. This is not an ideal estimate, but we found that it worked well in practice.

Check that the hypothesis class contains the ground truth. The algorithm described in [?] implicitly assumes that the hypothesis class \mathcal{Y} contains the ground truth labels S_k . This means that if S^+ is the maximizer of (21) then

$$\langle \mathbf{w}, \Psi(\Theta_k, S_k) \rangle - \langle \mathbf{w}, \Psi(\Theta_k, S^+) \rangle - \Delta(S^+, S_k) \leq 0, \quad (29)$$

since otherwise we would have

$$\langle \mathbf{w}, \Psi(\Theta_k, S_k) \rangle + \Delta(S_k, S_k) > \langle \mathbf{w}, \Psi(\Theta_k, S^+) \rangle + \Delta(S^+, S_k), \quad (30)$$

contradicting S^+ as the maximizer of (21). However, our output space contains fundamentally real-valued quantities such as polygon vertices, which are recovered only to some finite precision by the inference algorithm, and since our ground truth labels were acquired by manual labelling, they sometimes exceed the maximum precision of the inference algorithm. In this case we effectively have $S_k \notin \mathcal{Y}$ (although there is always some $S' \in \mathcal{Y}$ close to S_k), so it is possible that S^+ violates (29). Our workaround here is simply to check the condition (29) each time we solve the separation problem and, if violated, substitute S_k for S^+ . This is justified by the observation that if (29) is violated for S^+ then it is violated for all $S \in \mathcal{Y}$. One could think of this as learning with respect to the hypothesis class $\mathcal{Y} \cup \{S_k\}$ but evaluating with respect to \mathcal{Y} . This is not an ideal solution but we found it to work well in practice. Unfortunately this patch has the side-effect of hiding bugs in the inference algorithm, so care is warranted.

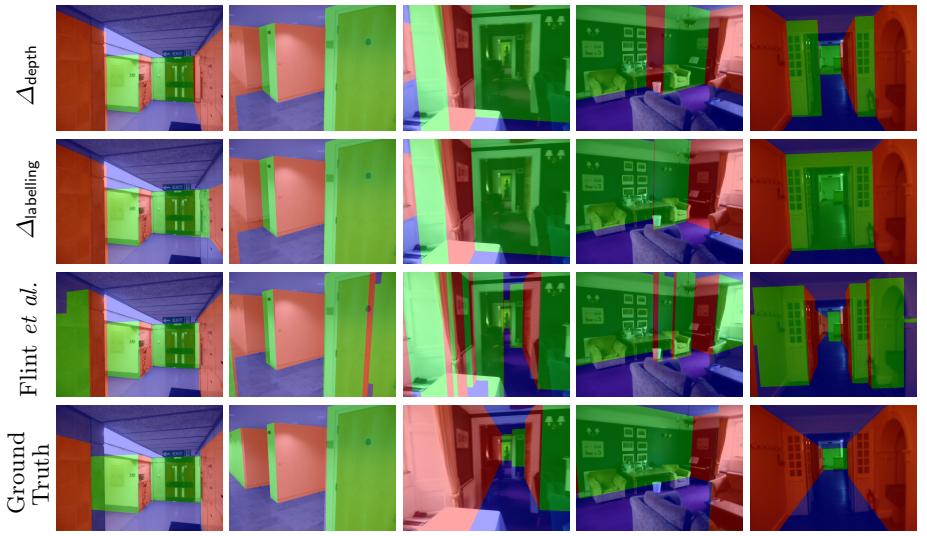


Fig. 7: Multiple-view reconstructions predicted by our system (held-out samples). The first two rows represent the predictors trained on Δ_{depth} and $\Delta_{\text{labelling}}$ respectively, the third row is from [?], and the fourth row is ground truth.

8 Conclusion

We have presented a unified learning framework for reconstructing polygonal models from single and multiple views of a scene. We have chosen to work with the indoor Manhattan class of models in order to leverage the parametrisation and inference algorithm recently proposed for this hypothesis class [?, ?]. Our approach to learning performs a single optimisation with respect to a clearly defined loss function. Experiments show our system out-performing the state-of-the-art for multiple-view reconstruction (by a large margin) and on one metric for single-view reconstruction.

In future work will extend this approach to learn geometry together with scene classifiers and context-aware object detectors, optimising with respect to a single joint loss function.