Physics Inspired Optimization on Semantic Transfer Features: An Alternative Method for Room Layout Estimation

Hao Zhao[1]*, Ming Lu[1], Anbang Yao[2], Yiwen Guo[2], Yurong Chen[2], Li Zhang[1]
[1]Department of Electronic Engineering, Tsinghua University
[2]Cognitive Computing Laboratory, Intel Labs China
{zhao-h13@mails,lu-m13@mails,chinazhangli@mail}.tsighua.edu.cn
{anbang.yao, yiwen.guo, yurong.chen}@intel.com

Abstract

In this paper, we propose an alternative method to estimate room layouts of cluttered indoor scenes. This method enjoys the benefits of two novel techniques. The first one is ***semantic transfer*** *(ST), which is: (1) a formulation to integrate the relationship between scene clutter and room layout into convolutional neural networks; (2) an architecture that can be end-to-end trained; (3) a practical strategy to initialize weights for very deep networks under unbalanced training data distribution. ST allows us to extract highly robust features under various circumstances, and in order to address the computation redundance hidden in these features we develop a principled and efficient inference scheme named* ***physics inspired optimization*** *(PIO). PIO's basic idea is to formulate some phenomena observed in ST features into mechanics concepts. Evaluations on public datasets LSUN and Hedau show that the proposed method is more accurate than state-of-the-art methods.*

1. Introduction

Given an input RGB image, a room layout estimation algorithm should output all the wall-floor, wall-wall, and wall-ceiling edges (depicted by Fig 1). This is a fundamental indoor scene understanding task as it can provide a strong prior for other tasks like depth recovery from a single RGB image [7][6] or indoor object pose estimation [23][9][22]. Besides, the room layout itself provides a high-level representation of an indoor scene for emerging applications like intelligent robots and augmented reality. This problem draws constant attention since the publication of the seminal work [11], and there are two lines of followers:

(1) As the upper part of Fig 1 shows, conventional methods follow a *proposing-ranking* scheme. Typically, the *proposing* part consists of three sub-modules as edge detection, vanishing point voting and ray sampling. With hand-crafted features and structured inference techniques, the *ranking* part outputs the best layout proposal, sometimes along with a representation of the clutter.

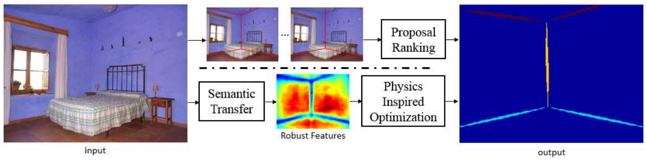

Figure 1. Above is the overview of conventional methods. Below is the overview of our method. Better viewed electronically.

(2) Recent methods [17][3][27] achieve dramatic performance improvements via features produced by fully convolutional networks (FCNs). [17][27] still follow the traditional *proposing-ranking* scheme. [3] is a proposal-free solution in which all those steps about proposal generation are eliminated. And instead of proposal ranking, in [3] inference is achieved through an optimization module.

Alternative to these two lines of works, we propose a method that features the advantages of both yet goes beyond them. It is illustrated by the lower part of Fig 1 and the motivations are in two folds:

Conventional methods. They provide many useful insights about indoor scene understanding. [11] and its followers [25][21][20][4][5][29] explore different ways to model the relationship between room layout and scene clutter. This effort is reasonable because the major challenges of room layout estimation lie here. Take Fig 1 for example, over 50% of wall-floor edge pixels is occluded by the bed. If the bed does not exist, this task will become much easier. However, these insights are not visited in recent FCN-based room layout estimation works. When designing networks, they treat FCNs as black boxes, taking no scene clutter information into consideration. As modelling

*This work was done when Hao Zhao was an intern at Intel Labs China supervised by Anbang Yao who is responsible for correspondence.

meaningful concepts with a neural network has always been difficult, it motivates us to explore the possibility to describe scene clutter within an FCN.

FCN-based methods. Unlike [17][27] which still follow the *proposing-ranking* scheme, [3]'s framework shows intriguing compactness. However, its optimization is primitive in which the sampled solution space is exhaustively searched and no gradient is modeled. So the second motivation of this paper is to develop a principled, gradient-based, and efficient optimization algorithm for this task.

Guided by the first motivation, we propose **Semantic transfer** (ST) which has three features from three different perspectives: 1) As a discriminative model, it integrates the relationship between room layout and scene clutter into an FCN. 2) As an architecture, it enjoys the benefit of end-to-end training. 3) As a training strategy, it provides better network initialization and allows us to train a very deep network under unbalanced training data distribution. ST provides highly robust features under various circumstances. Accordingly we propose an inference technique named **Physics inspired optimization** (PIO). ST and PIO play different yet closely interdependent roles because the core idea of PIO is to formulate some phenomena observed in ST feature maps with mechanics concepts.

2. Related Works

Conventional methods. The standard definition of room layout estimation is firstly introduced by [11]. It clusters edges into lines joining at three vanishing points, according to the famous Manhattan assumption [2]. Then a lot of layout proposals are generated by ray sampling. Hand-crafted features are used to learn a regressor for proposal ranking. Later on, many works try to improve this framework. [19] detects conjunctions instead of edges and modifies proposal generation and ranking accordingly. While ranking room layouts, [25] simultaneously estimates a clutter mask. [21] aims to improve the inference efficiency of methods like [25]. Going beyond estimating clutter mask, [20] estimates objects' 3D bounding boxes and room layout during inference. Except for learnt clutter representations, [4] incorporates furniture shape prior. In [5]'s formulation, furniture is modeled with parts instead of a box. [29] goes even further by modelling furniture relationship with scene grammars.

FCN-based methods. Recently [17] trains an FCN for pixel-wise edge labelling, with every pixel assigned a label from this 4-class set S {background (bg), wall-floor edge (wf), wall-wall edge (ww), wall-ceiling edge (wc)}.

$$S = \{bg, wf, ww, wc\} \tag{1}$$

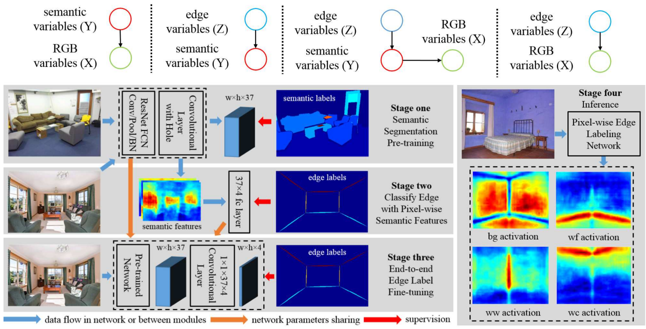

Figure 2. Top: Probabilistic node connectivity. Bottom: Semantic transfer. In stage three, *pre-trained network* refers to the one outlined by the dashed box in stage one. In stage four, *pixel-wise edge labelling network* refers to the one outlined by the dashed box in stage three. Better viewed electronically for a higher resolution.

Activations of the last layer are incorporated into the conventional inference framework as features. [3] uses another formulation in which every pixel may be assigned a label from a 5-class set {floor, left wall, middle wall, right wall, ceiling}. This 5-class formulation has an ambiguity problem as the patterns of three type of walls are not discriminative in nature. FCN is coordinate-invariant since convolutional layers actually conduct a sliding window search, so it is not suitable to tell the difference between *left wall* and *right wall*. Thus [3] uses an additional ambiguity clarification step. [27] uses both formulations for FCN training. These FCN-based works show dramatic performance improvements but as stated by the second motivation, their inference schemes remain conventional or primitive. With robust FCN features, it is possible to design more principled and efficient inference schemes.

Broader literature. There are actually other scene understanding tasks substantially same as or similar to room layout estimation. For example, [18] tries to understand the layouts of natural scenes with a horizon, urban scenes, corridors and others, for which room layout estimation is only a special case. Another special case of [18] is *outdoor urban layout estimation*, such as [1][13]. It is often regarded as a graphics application under the name of *photo pop-up* and evaluated with subjective user study. [14] tries to recover more detailed room layout than a box and evaluates with wall-floor edge error. Since these works exploit techniques that [11] is built upon, they could potentially benefit from the method proposed in this paper.

Concepts similar to ST and PIO. If we look at an even broader literature, the concepts somewhat similar to ST and PIO have already been discussed. Under the name of label transfer, [16][28] address semantic segmentation in a non-parametric manner. ST is different from them primarily as a unified deep architecture (and of course in its parametric nature). [8] and its followers are famous for stating human limbs as springs. PIO is different from them primarily as an efficient approximation inspired by mechanics concepts.

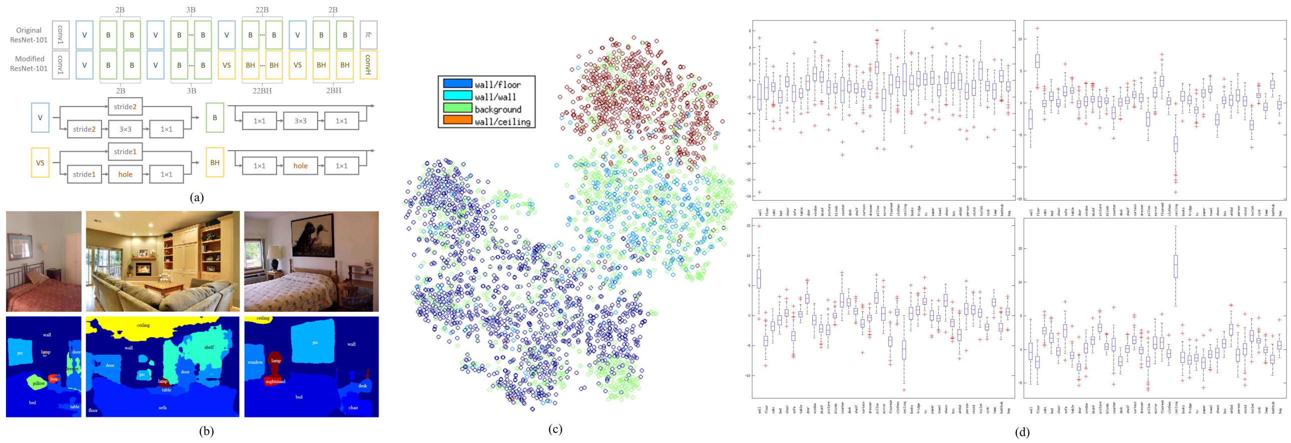

Figure 3. (a) Network design for ST stage one. (b) Qualitative results for semantic segmentation on dataset LSUN. Note that LSUN does not provide semantic segmentation ground truth. (c) Unsupervised structure visualization of the semantic feature space. (d) Transfer weights Visualization. Left-top: bg. Right-top: wf. Left-bottom: ww. Right-bottom: wc. Better viewed electronically for a higher resolution.

3. Semantic Transfer

Here we present semantic transfer which is made up of 3 stages (Fig 2). Firstly we look at the inference phase: the ultimate goal of our FCN is pixel-wise edge labelling. As demonstrated by Fig 2's *stage four* panel, four pixel-wise activation maps are extracted from the input image, with each one corresponding to a label from S (set 1). For example, in the *wf activation* map higher color temperature indicates higher possibility of *wf* existence.

In stage one, we train an FCN for 37-class semantic segmentation on dataset SUNRGBD in order to describe a cluttered scene to the utmost extent. These 37 categories can cover most of the stuff and furniture that commonly appear in an indoor scene, like wall, ceiling, chair or window. We build this FCN upon the newly introduced architecture ResNet-101 [10]. As Fig 3a shows, we do net surgeries to the last two sets of bottlenecks in original ResNet-101, with the *hole* mechanism described in [15] (under the name of dilated convolution in [26]). Inputs to this network (RGB images) are actually random variables X taking values from $[0, 255]$. X is determined by hidden random variables Y taking values from semantic labels $[1, 37]$. Thus this network describes the posterior distribution $P(Y|X)$.

In stage two, we feed the room layout dataset LSUN through the semantic segmentation network, producing pixel-wise 37-channel semantic features. Since they are both indoor scene understanding datasets, the model trained on SUNRGBD generalizes well on LSUN. Fig 3b shows some qualitative results on LSUN, all of which are produced by a softmax operation without post-processing techniques like conditional random fields. Then treating every pixel as a sample, we learn a fully connected layer to bridge the gap between 37-channel semantic features and 4-class edge labels. In order to illustrate that semantic features are discriminative for this task, we do a standard unsupervised analysis with t-sne [24]. As Fig 3c shows, samples of wall-ceiling edges (wc) and wall-floor edges (wf) form obvious clusters in the embedding space. Yet some samples of wall-wall edges (ww) and background (bg) scatter among each other. In this stage, Y is determined by hidden random varibles Z taking values from edge labels $[1, 4]$ (set 1). So this fc layer describes the posterior distribution $P(Z|Y)$.

$P(Z|Y)$ is a parameterized representation of the relationship between room layout and scene clutter. Unlike pioneering works, we model this relationship directly in a neural network. This is inspired by how a human understands room layout. As demonstrated by Fig 2's *stage two* panel, the network in stage one extracts 37-channel semantic features from the scene. Only the channel on top of the stack is fully illustrated, and that channel corresponds to *window*. This channel can roughly tell the locations and extensions of three windows in the scene. How would a human brain parse room layout from semantic features like this? We hypothesize that it makes decisions according to rules like:

wall-floor edges cannot go through windows, so they are less possible to appear in areas with high window scores.

In order to validate that the network behaviors are consistent to this hypothesis, we visualize the transfer weights in this fc layer. These weights are learnt for 100 times independently and organized into box figure as Fig 3d. Not surprisingly, *wall*, *floor* and *ceiling* channels of semantic features contribute the most to *ww*, *wf* and *wc*, respectively. Generally speaking, higher scores with smaller boxes means stronger correlation. We take *wc* for example. Except for *ceiling*, top four transfer weights come with *cabinet*, *picture*, *sofa* and *whiteboard*. According to common sense, *cabinet*, *picture* and *whiteboard* tend to appear in the receptive field of a *wc* pixel, because they are vertically high in physical space. *sofa* is usually lower, so its variation (depicted by box size) is twice *picture*'s. *whiteboard* is rare, explaining why its variation is also large.

In stage three, this learnt 37×4 fc layer is reshaped into a $1 \times 1 \times 37 \times 4$ convolutional layer and added on top of the network trained in stage one. Weights from stage one act as a feature extractor, and weights from stage two act as a classifier. They form a pixel-wise edge labelling network describing $P(Z|Y)P(Y|X) = P(Z|X)$. On one hand, this network can be end-to-end fine-tuned on LSUN for edge labelling, which is the ultimate goal we mentioned at the beginning of this section. On the other hand, it incorporates the relationship between scene clutter and room layout elegantly, which is the first motivation of this paper.

Except for end-to-end training and scene clutter modelling, another advantage of semantic transfer is better initialization for extremely unbalanced training data. We have tried to train this pixel-wise edge labelling network directly with the ResNet FCN (Fig 3a), leaving out ST. But outputs of the batch normalization (BN) layers are prone to overflow, making training fail. Training problems are also reported in [17]'s 3.2 section. It says the network has to be pre-trained on NYUd2 and pre-training on PASCAL leads to bad results. This problem may be caused by the extremely unbalanced distribution of edge labels. As shown by Fig 2's *stage two* panel, over 99% labels are *background*. Like the classical method of initializing an auto-encoder with multiple restricted boltzmann machines [12], our pixel-wise edge labelling network is initialized by the first two stages. We no longer observe the overflow phenomenon with ST.

Probabilistic nodes' connectivity is illustrated by Fig 2's upper part. The pre-trained model will be released. Details about network, unsupervised analysis and weights visualization are provided in the supplementary material.

4. Optimization

We provide comprehensive feature quality visualizations and comparisons in the supplementary material. For parameterized room layout inference, we propose two techniques: naive optimization (NO) and its efficient approximation named physics inspired optimization (PIO).

There are 11 different possible room layout topologies in a 2D image, as demonstrated in the supplementary material. We index them with i. Each topology is parameterized by the edge conjunctions set $P_i = \{P_{ij}, j \in [1, nC]\}$, with every P_{ij} as a 2D coordinate and nC as the conjunction number. Conjunction connectivity is defined by edge set $E_i = \{E_{ik} = (Q_{ka}, Q_{kb}, c), Q_{ka} \in P_i, Q_{kb} \in P_i, c \in S, k \in [1, nE]\}$, with nE as the edge number. S is set 1. The 6th topology is demonstrated in Fig 4 as an example. P_i and E_i can be converted into a pixel-wise edge label map which is similar to the output in Fig 1. This **conversion** is denoted as $M = C(P_i, E_i)$ and we will omit E_i later because it does not change for a certain topology. Also, we will use $M[P_i]$ when referring the map M produced from conjunction set P_i and $M[P_{ij}]$ when a certain conjunction P_{ij} is under consideration.

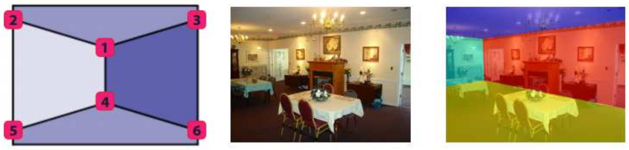

Figure 4. The 6th topology, clipped from LSUN specification.

The features produced by the pixel-wise edge labelling network are denoted as $F_l (l \in [1, 4])$. Note that both M and F_l are of the same size as input image, denoted by (w, h). On them we define the consistency objective (CO) and its corresponding energy format (e):

$$CO = \frac{1}{wh} \sum_{l=1}^{4} \sum_{m=1}^{w} \sum_{n=1}^{h} F_l(m, n) \times M_l(m, n) \quad (2)$$

$$e = \exp(-CO) \quad (3)$$

in which $M_l (l \in [1, 4])$ is the binary mask generated from M by setting a pixel to one if $M(m, n) = l$ and zero otherwise. For every different topology we can find the best parameterized representation P_i by minimizing e:

$$\overline{P_i} = \arg\min_{P_i} e \quad (4)$$

In most cases, starting from the right topology leads to the lowest energy value and wrong topologies lead to higher energy values. Failure cases do exist and we will visualize them later. All optimization implementations detailed below are initialized from the average state of P_i set (such as the one demonstrated by Fig 4's left figure).

4.1. Naive Optimization

To solve Equation 4, firstly we propose NO as follows:

$$\frac{\partial e}{\partial P_{ijx}} \approx e(P_{ij(x+\Delta x)}) - e(P_{ij(x-\Delta x)}) \quad (5)$$

$$\frac{\partial e}{\partial P_{ijy}} \approx e(P_{ij(y+\Delta y)}) - e(P_{ij(y-\Delta y)}) \quad (6)$$

$$\Delta P_{ij} = \alpha \times (-\frac{\partial e}{\partial P_{ijx}}, -\frac{\partial e}{\partial P_{ijy}}) \quad (7)$$

in which α is the scaling factor and Δx $(= \Delta y)$ is the window size. For conjunctions at image boundary (e.g. P_{62}

Algorithm 1 Naive Optimization

```
Initialize: average $P_i$
while $e$ decreases do
    for all $j$ do
        update $P_{ij}$ according to Equation 5, 6, and 7
    end for
    calculate $e$ at updated $P_i$
end while
```

in Fig 4's left figure), an additional constraint is imposed by setting corresponding component of ΔP_{ij} to zero. If conjunctions move to image corners, ΔP_{ij} is treated as a special case so as to allow the conjunction to move onto another boundary or just stick to the corner. The convergence performance of NO is good but it is very slow so we introduce PIO as an efficient alternative.

4.2. Analysis and Motivation

We first analyze the efficiency bottleneck of NO. When calculating Equation 5 (and similarly Equation 6),

$$\frac{\partial e}{\partial P_{ijx}} \propto -(CO(P_{ij(x+\Delta x)}) - CO(P_{ij(x-\Delta x)})) \qquad (8)$$

$$= -\sum_{l=1}^{4}\sum_{m,n} F_l \times (M_l[P_{ij(x+\Delta x)}] - M_l[P_{ij(x-\Delta x)}]) \quad (9)$$

In Equation 9, we omit m, n whose meanings are stated in Equation 2. Calculating $M_l[P_{ij(x+\Delta x)}] - M_l[P_{ij(x-\Delta x)}]$ is the efficiency bottleneck, which represents $M = C(P_i)$'s gradient and we illustrate $M_2[P_{ij(x+\Delta x)}] - M_2[P_{ij(x-\Delta x)}]$ with Fig 5a, which subtracts two pixel-wise masks.

As a reminder, M is generated by conversion C. At first, we implement C by traversing every pixel to decide its label. If we denote the scale of w, h by N, the complexity of this implementation (referred as NOA later) is $O(N^2)$ for every calculation of Equation 5 or 6. It runs for tens of minutes for an image. An improved implementation (referred as NOB later) of C calculates pixel coordinates between two conjunctions and accesses corresponding mask element directly. Its complexity is $O(N)$, and it runs for about 30s for an image. The idea of further reducing the complexity to $O(1)$ motivates us to introduce PIO.

We consider every edge as a spring which may translate, rotate and change its length. In NO, edges' movements are decided by every pixel on them, yet there is computation redundance. As demonstrated by Fig 5c and Fig 5d, we consider the feature map as a potential field and analyze how points on the edge move. Not surprisingly, their movements are not independent and can be roughly interpolated from the movements of the edge's two endpoints, that is Q_{ka} and Q_{kb}. Based on this observation, we propose to approximate ΔP_{ij} with gradients defined on P_i instead of $M_l[P_i]$. Since the number nC of conjunctions P_i is constant, the complexity is $O(1)$. This is PIO's first key concept.

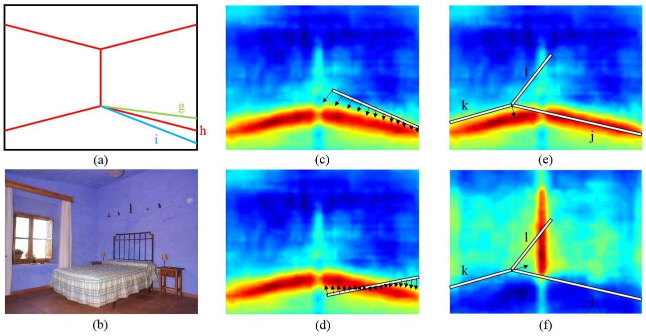

Figure 5. (ag) $M_2[P_{ij(x-\Delta x)}]$. (ah) $M_2[P_{ij}]$. (ai) $M_2[P_{ij(x+\Delta x)}]$. (b) Input image. (c/d) If we consider an edge as a spring and the feature map as a potential field, forces imposed on the spring's every point are correlated. (e/f) The influence of force composition.

Force composition is the second key concept of PIO. As demonstrated by Fig 5e, if we consider the endpoints of edge j and k instead of every points on them they will move towards a local minima state. This will be corrected by calculating the movements of edge l (Fig 5f), in which another feature map (wall-wall edge) will be used as the potential field. So the movement of every conjunction should be decided by the forces imposed on every edge that is connected to that conjunction. Obviously adding two gradient vectors (such as the ones in Fig 5e and Fig 5f) naturally obeys the parallelogram law of force composition.

4.3. Physics Inspired Optimization

For the first concept, we define a new consistency objective for each endpoint of an edge $E_{ik} = (Q_{ka}, Q_{kb}, c)$:

$$CO2 = F_c(Q_{kax}, Q_{kay}) \qquad (10)$$

$$e2 = \exp(-CO2) \qquad (11)$$

As a reminder, the meanings of E and F are stated at the beginning of this section. Calculating the gradient of a certain point in a potential field is trivial as:

$$\frac{\partial e2}{\partial Q_{kax}} \approx e2(Q_{ka(x+\Delta x)}) - e2(Q_{ka(x-\Delta x)}) \qquad (12)$$

$$\frac{\partial e2}{\partial Q_{kay}} \approx e2(Q_{ka(y+\Delta y)}) - e2(Q_{ka(y-\Delta y)}) \qquad (13)$$

$$\Delta Q_{ka} = \alpha \times (-\frac{\partial e2}{\partial Q_{kax}}, -\frac{\partial e2}{\partial Q_{kay}}) \qquad (14)$$

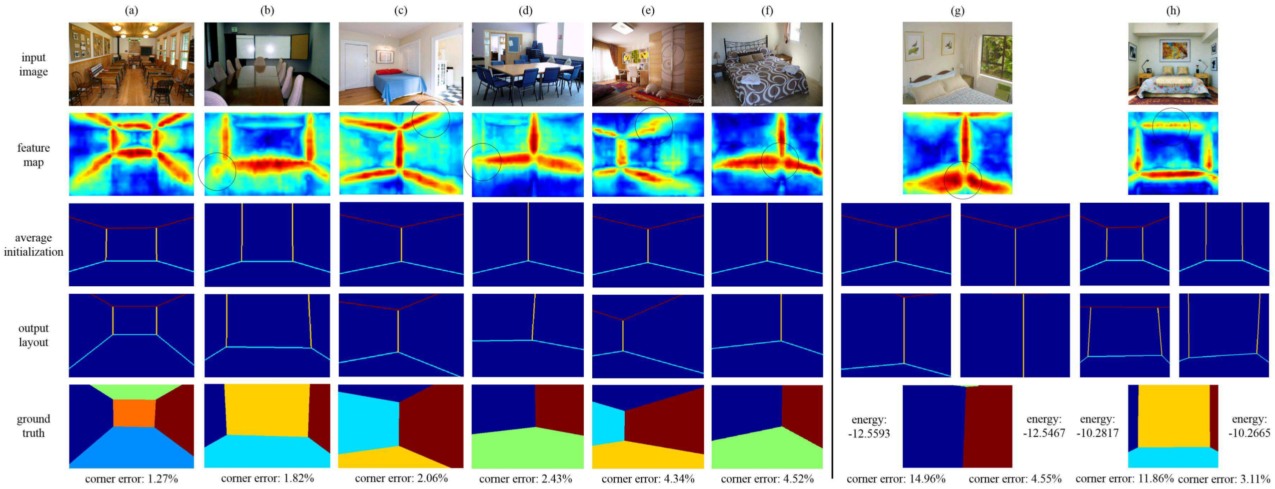

Figure 6. Left: qualitative results on LSUN validation set. The visualized feature map merges wf, ww, and wc by a pixel-wise max operation, yet they are used independently in PIO. Right: typical failure cases in which a wrong topology produces the lowest energy.

In this physics inspired optimization, ΔQ_{ka} is regarded as a force imposed on the endpoint Q_{ka} of an spring-like edge E_{ik}. As for the second concept of force composition, we define $E[P_{ij}] = \{(Q_{oa} = P_{ij}, Q_{ob}, c), o \in [1, \#(E[P_{ij}])]\}$ which is a subset of E_i. And the force imposed on P_{ij} when considering different edges can be denoted as ΔQ_{oa}. Thus we approximate ΔP_{ij} with:

$$\Delta P_{ij} = \sum_{o=1}^{\#(E[P_{ij}])} \Delta Q_{oa} \tag{15}$$

Algorithm 2 Physics Inspired Optimization

```
Initialize: average $P_i$
while $e$ decreases do
    for all $j$ do
        get the subset $E[P_{ij}]$
        for all $o$ do
            calculate the force imposed on $Q_{oa} = P_{ij}$ according to Equation 12 13 14
        end for
        calculate $\Delta P_{ij}$ by Equation 15, and update $P_{ij}$
    end for
    calculate $e$ at updated $P_i$
end while
```

As mentioned before, Equation 15 naturally obeys the parallelogram law of force composition. In case of potential confusion, we clarify that both ΔQ_{oa} and ΔQ_{ka} are calculated according to Equation 14 (k is the index in E_i and o is the index in E_i's subset $E[P_{ij}]$). And to summarize, PIO's efficiency primarily comes from Equation 10's $O(1)$ complexity while Equation 2's is at least $O(N)$ (NOB).

5. Experiments

5.1. LSUN Results

LSUN is a room layout estimation dataset consisted of 4000 training, 394 validation, and 1000 held-out testing samples. Two standard metrics are used for evaluation: (1) $\mathbf{e_{corner}}$. Corner (conjunction) error is the Euclidean distance between estimated coordinates of P_i and ground truth. Because of resolution diversity, $\mathbf{e_{corner}}$ is normalized by image diagonal length. (2) $\mathbf{e_{pixel}}$. By converting P_i into mask representation like the ground truth in Fig 6, pixel error measures the ratio of mislabelled pixels to all pixels. (For $\mathbf{e_{pixel}}$'s label ambiguity problem, LSUN official evaluation codes automatically maximize the overlap.)

For a large-scale evaluation, both metrics are averaged over images. On validation set, official evaluation codes provided by LSUN committee are used. Third-party evaluation results on the test set are reported in Table 1. The proposed method outperforms conventional method [11] and FCN-based methods [17][3][27] on both metrics. Qualitative results and failure cases on validation set are demonstrated by Fig 6. Eight videos showing how PIO works are provided in the supplementary material, with each one corresponding a sample in Fig 6, respectively.

Fig 6a shows a typical easy case, in which most edge pixels are visible and the feature map captures their locations accurately. As *video-a.wmv* shows, the visualized edge map gets twisted temporarily near 30th iteration because of force composition and PIO finally aligns it with the true layout.

Fig 6bcd show some cases in which the feature maps fail to locate edges accurately where the black circles are, leading to relatively higher $\mathbf{e_{corner}}$. Reasons are diverse, such as severe occlusion (b), insufficient feature map resolution (c),

Method	$\mathbf{e_{pixel}}$ (%)	$\mathbf{e_{corner}}$ (%)
Hedau et al.(2009) [11]	24.23	15.48
Mallya et al.(2015) [17]	16.71	11.02
Dasgupta et al.(2016) [3]	10.63	8.20
Ren et al.(2016) [27]	7.57	5.23
Ours	**5.29**	**3.84**

Table 1. Quantitative results on LSUN test set.

Method	$\mathbf{e_{pixel}}$ (%)
Hedau et al.(2009) [11]	21.20
Del Pero et al.(2013) [5]	12.70
Mallya et al.(2015) [17]	12.83
Dasgupta et al.(2016) [3]	9.73
Ren et al.(2016) [27]	8.67
Ours	**6.60**

Table 2. Quantitative results on Hedau test set. To clarify, [11][5] are not trained on the large-scale dataset LSUN.

and misleading texture (d).

In Fig 6e, the room is no longer a strict box if we consider the cabinet as a part of wall. Actually those separated wall-ceiling edges are successfully captured by the feature map and aligned by PIO. However, the annotation protocol takes the cabinet as occlusion. Fig 6f shows a heavily-occluded case. Semantic transfer allows the network to extrapolate the existence of wall-floor edges behind the bed, but the conjunction in the black circle is not accurately localized.

Although not 100% accurate, Fig 6a-f are regarded as successful cases as the output topologies are right. Fig 6gh are two typical failure cases in which a wrong topology produces the lowest energy. Fig 6g's failure is caused by overfitting. The network extrapolates there are wall-floor edges behind the bed, yet the annotation protocol does not. *video-g1.wmv* shows the optimization procedure of the wrong topology and *video-g2.wmv* shows that of the right topology. Even though the latter leads to a lower error but the algorithm outputs the former as it produces a lower energy. Fig 6h demonstrates another type of failures caused by structure ambiguity. Again this scene is no longer a strict box as some parts of the wall protrude outwards. The network recognizes them as ceiling but the annotation protocol does not, causing PIO to output a wrong topology.

5.2. Hedau Results

The Hedau dataset is presented by [11], being consisted of 209 training samples and 105 testing samples. On Hedau test set, We directly evaluate the model trained with LSUN training set. As Fig 7 shows, this model extracts reliable features across datasets. Consistent to the literature we use pixel error as quantitative metric. We report better results than conventional methods like [11][5] and FCN-based methods like [17][3][27] (Table 2). Overall pixel error (6.60%) on Hedau test set is higher than that (5.29%) on LSUN test set because the ground truth mask annotated by Hedau dataset is more strict (typically shown by Fig 7j).

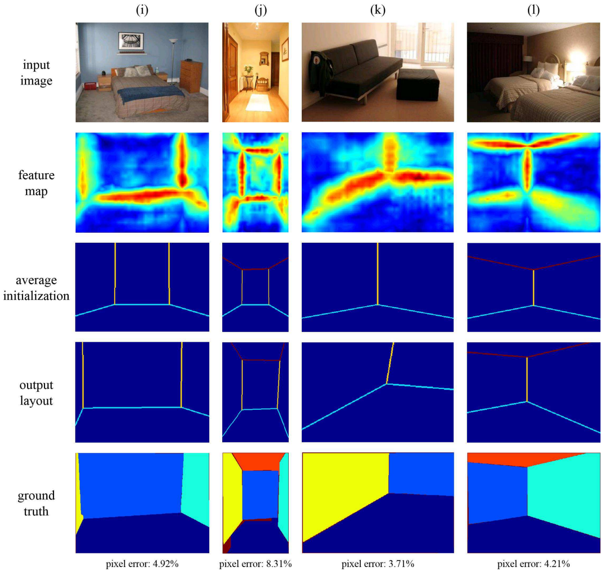

Figure 7. Qualitative results on Hedau test set.

	NOB	PIO
artpf (s)	35.41	1.79
$\mathbf{e_{pixel}}$ (%)	5.42	5.48
$\mathbf{e_{corner}}$ (%)	3.88	3.95

Table 3. Average running time per frame (artpf) comparison.

5.3. Hyper Parameters and Efficiency

(1) In Algorithm 2, wether e decreases is determined by a threshold of 10^{-6}. This threshold is related to the numerical scale of e. During implementation, we use $e = -CO$ instead of $e = \exp(-CO)$ because of equivalence, and e's numerical scale is around -15 which has already been shown in videos mentioned above. (2) Scaling factor α is self-adaptive to ensure that the gradients' (forces') length is between 1 and 3. This restricts the conjunction to move only a little in one iteration, as the videos show. (3) The influence of window size Δx $(= \Delta y)$ is evaluated on LSUN validation set, and the quantitative results are demonstrated by Fig 8. As the window size grows from 1 pixel to 10 pixels, both metrics show a trend of increase. Since PIO can be regarded as an alignment algorithm, this is not surprising because calculating gradients in a larger window size leads to a weaker ability to accurately capture local structure.

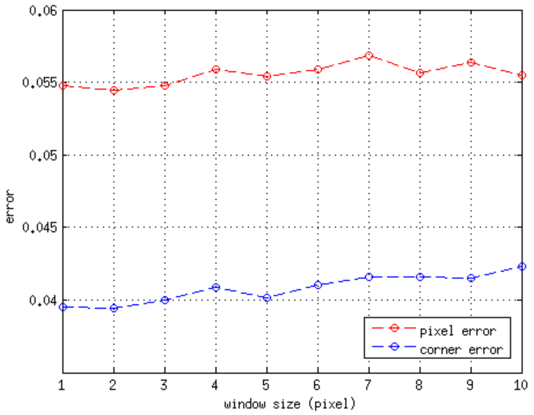

Figure 8. LSUN validation set error v.s. window size.

We evaluate average running time per frame on LSUN validation set with NOB and PIO. NOA is not evaluated because of intractability. The results are provided in Table 3, showing that PIO brings dramatic speedup against NOB without causing noticeable accuracy loss. Our codes are all implemented with MATLAB, thus there is still much head room for potential real-time applications.

5.4. Ablative Study of ST

In order to evaluate the impact of semantic transfer, we use three standard edge prediction accuracy metrics as follows: F-score @ optimal dataset scale (ODS), F-score @ optimal image scale (OIS), average precision (AP). We consider following settings: (A) directly train a VGG16-based network for edge labelling. (B) train a VGG16-based network with semantic transfer. (C) train a Resnet101-based network with semantic transfer. All settings (including D-G in next subsection) use same hyperparameters. As shown by Table 4, setting B comes with a higher accuracy than A (2.8%ΔODS), and setting C sees a larger improvement (4.3%ΔODS). This indicates that both semantic transfer and the introduction of Resnet101 bring improvements yet the latter takes a relatively larger part.

5.5. Representation Learning Perspective

In order to further compare semantic transfer with traditional representation learning schemes, we consider these settings (all on a VGG16-based network): (D) pre-train on SUNRGBD for semantic segmentation, re-initialize the last layer and fine-tune all parameters. (E) same as D except that we only fine-tune parameters after layer 5b. (F) semantic transfer and in stage three fine-tune all parameters (It is same as B). (G) same as F except that we only fine-tune parameters after layer 5b. As shown by Table 4, F's performance is slightly better than D (0.3%ΔODS, one may argue this could be caused by additional parameters or stochastic training) yet this margin gets more significant when we freeze parameters before layer 5b (3.1%ΔODS comparing G against E). This indicates that tuning a (properly initialized) 37 × 4 transfer layer is easier than re-training a (gaussian initialized) classification layer (more obvious in the frozen representation settings).

	A	B(F)	C	D	E	G
ODS	0.243	0.271	**0.314**	0.268	0.202	0.233
OIS	0.251	0.285	**0.328**	0.280	0.208	0.236
AP	0.135	0.151	**0.184**	0.148	0.091	0.098

Table 4. Ablative study of ST on LSUN validation set.

	H	I	J	K
$\mathbf{e}_{\textbf{pixel}}$	11.28	6.31	5.75	**5.48**
$\mathbf{e}_{\textbf{corner}}$	8.55	4.98	4.17	**3.95**

Table 5. Ablative study of PIO on LSUN validation set.

5.6. Ablative Study of PIO

With the classical pipeline (edge detection, vanishing point voting, ray sampling), we extract on average 334 proposals per image on LSUN validation set. Then with semantic transfer features (setting C above), we consider these settings: (H) pick the proposal that correlates to the features the most. (I) do PIO with the best proposal. (J) do PIO with top 10 best proposals and pick the one with the lowest energy. (K) the PIO setting mentioned above (without depending on those error-prone proposals generated from low-level edge cues). I sees a higher accuracy than H (-4.97%$\Delta\mathbf{e}_{\textbf{pixel}}$), which is not surprising as PIO refines layout proposals. Yet since I is restricted by the the proposal quality (which degenerates heavily in highly-occluded cases) thus K outperforms I (-0.83%$\Delta\mathbf{e}_{\textbf{pixel}}$). Augmenting with 10 proposals (J) sees a comparable performance with K. And generally speaking, PIO is better than ranking proposals (-5.80%$\Delta\mathbf{e}_{\textbf{pixel}}$ comparing K and H).

6. Conclusion

In this paper, we propose an alternative method for room layout estimation. With a very deep semantic transfer FCN, we extract reliable edge features under various circumstances. Meanwhile we develop PIO as a new inference scheme, which is inspired by mechanics concepts. The method's effectiveness is illustrated by extensive quantitative experiments on public datasets. Figures and videos are also provided as intuitive demonstrations.

Acknowledgements. This work was jointly supported by National Natural Science Foundation of China (Grant No.61132007, 61172125, 61601021, and U1533132).

References

[1] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, and A. Konushin. Fast automatic single-view 3-d reconstruction of urban scenes. In *ECCV 2008*.

[2] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *ICCV 1999*.

[3] S. Dasgupta, K. Fang, K. Chen, and S. Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *CVPR 2016*.

[4] L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *CVPR 2012*.

[5] L. Del Pero, J. Bowdish, B. Kermgard, E. Hartley, and K. Barnard. Understanding bayesian rooms using composite 3d object models. In *CVPR 2013*.

[6] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *CVPR 2015*.

[7] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS 2014*.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV 2005*.

[9] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *CVPR 2015*.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR 2016*.

[11] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *CVPR 2009*.

[12] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. In *Science 2006*.

[13] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV 2007*.

[14] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR 2009*.

[15] C. Liang-Chieh, P. George, K. Iasonas, M. Kevin, and L. Y. Alan. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR 2015*.

[16] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR 2009*.

[17] A. Mallya and S. Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *ICCV 2015*.

[18] V. Nedovic, A. W. Smeulders, A. Redert, and J.-M. Geusebroek. Depth information by stage classification. In *ICCV 2007*.

[19] S. Ramalingam, J. K. Pillai, A. Jain, and Y. Taguchi. Manhattan junction catalogue for spatial reasoning of indoor scenes. In *CVPR 2013*.

[20] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3d layout and object reasoning from single images. In *ICCV 2013*.

[21] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3d indoor scene understanding. In *CVPR 2012*.

[22] S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *CVPR 2016*.

[23] S. Song and J. Xiao. Sliding shapes for 3d object detection in depth images. In *ECCV 2014*.

[24] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research 2008*.

[25] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. *ECCV 2010*.

[26] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR 2016*.

[27] R. Yuzhuo, C. Chen, L. Shangwen, and K. C.-C. Jay. A coarse-to-fine indoor layout estimation (cfile) method. In *ACCV 2016*.

[28] H. Zhang, J. Xiao, and L. Quan. Supervised label transfer for semantic segmentation of street scenes. In *ECCV 2010*.

[29] Y. Zhao and S.-C. Zhu. Scene parsing by integrating function, geometry and appearance models. In *CVPR 2013*.