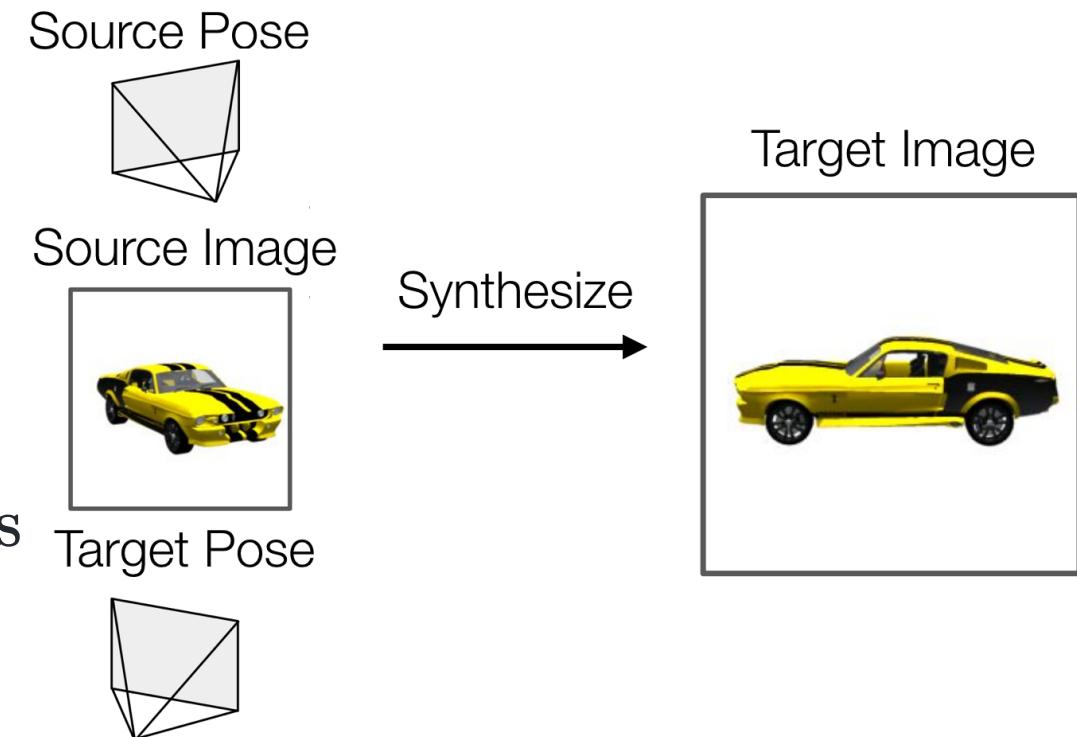
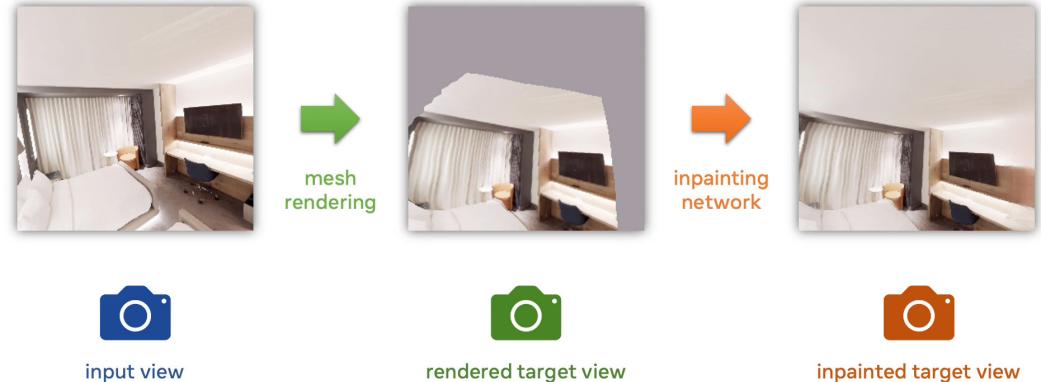


Multi-Plane Program Induction with 3D Box Priors

Yikai Li*, Jiayuan Mao*, Xiuming Zhang, William T. Freeman, Joshua B. Tenenbaum,
Noah Snavely, Jiajun Wu

Motivation

- Task: Single Image View Synthesis
- Challenge: Learning scene decompositions is, generally, underspecified.
- Observation: Physical objects display a measure of uniformity.
- Hypothesis: Can we impose a structural prior to capture this uniformity?
- Use a "box prior"
 - Pinhole camera
 - \exists 4 planes if inside else 2 planes
 - Each plane has repeating pattern.



Sample Box Programs

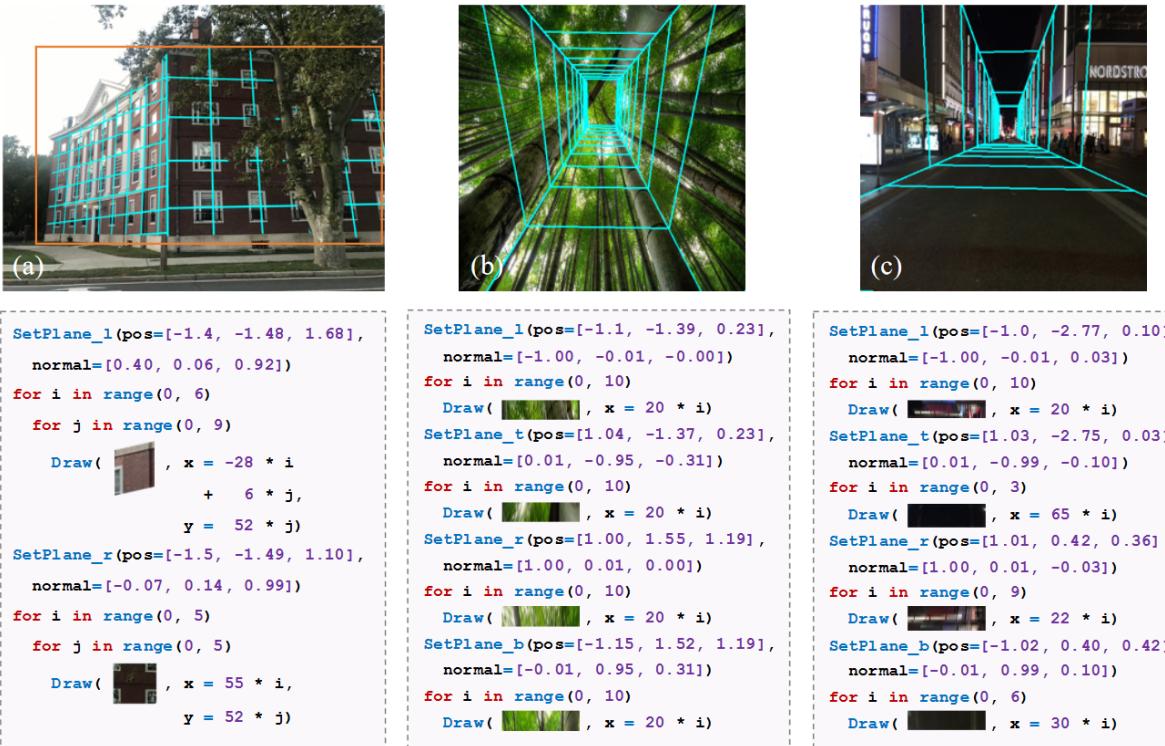


Figure 11: More image examples and the corresponding programs. We use cyan lines to visualize the lattice structure on each plane.

Program → CameraProgram; WorldProgram;
 CameraProgram → SetCamera(pos=Vec3, point_to=Vec3);
 WorldProgram → PlaneProgram; | PlaneProgram; World Program;
 PlaneProgram → SetPlane(pos=Vec3, normal=Vec3) For1Stmt;
 For1Stmt → For (i in range(Integer, Integer)) { DrawStmt; | For2Stmt }
 For2Stmt → For (j in range(Integer, Integer)) { DrawStmt }
 DrawStmt → Draw (x=Expr, y=Expr)
 Expr → Real × i + Real × j | Real × i

Table 1: The domain-specific language (DSL) of box programs. Language tokens **For**, **If**, **Integer**, **Real**, and arithmetic/logical operators follow the Python convention. **Vec3** denotes 3D real vectors.

Algorithm

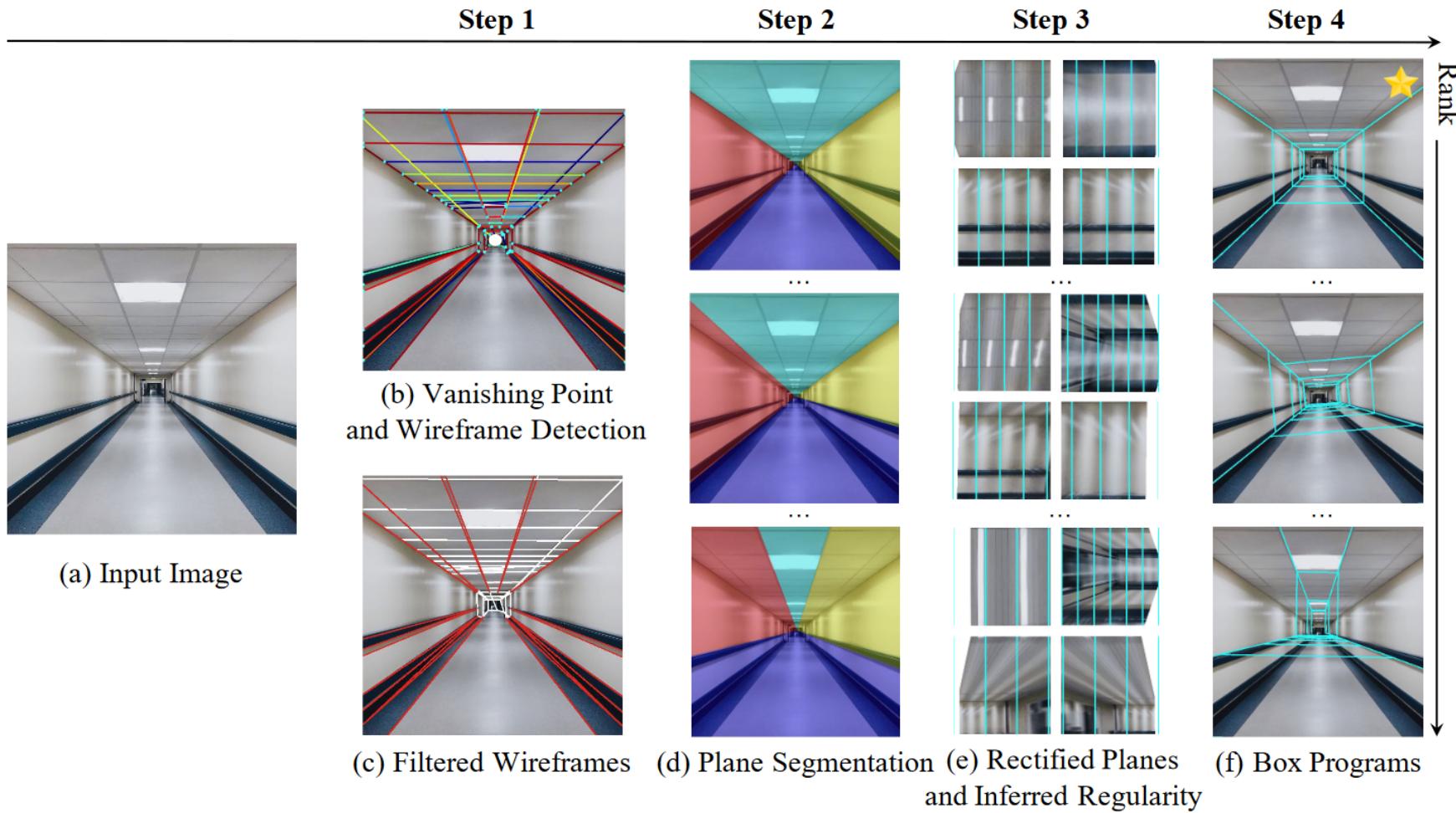
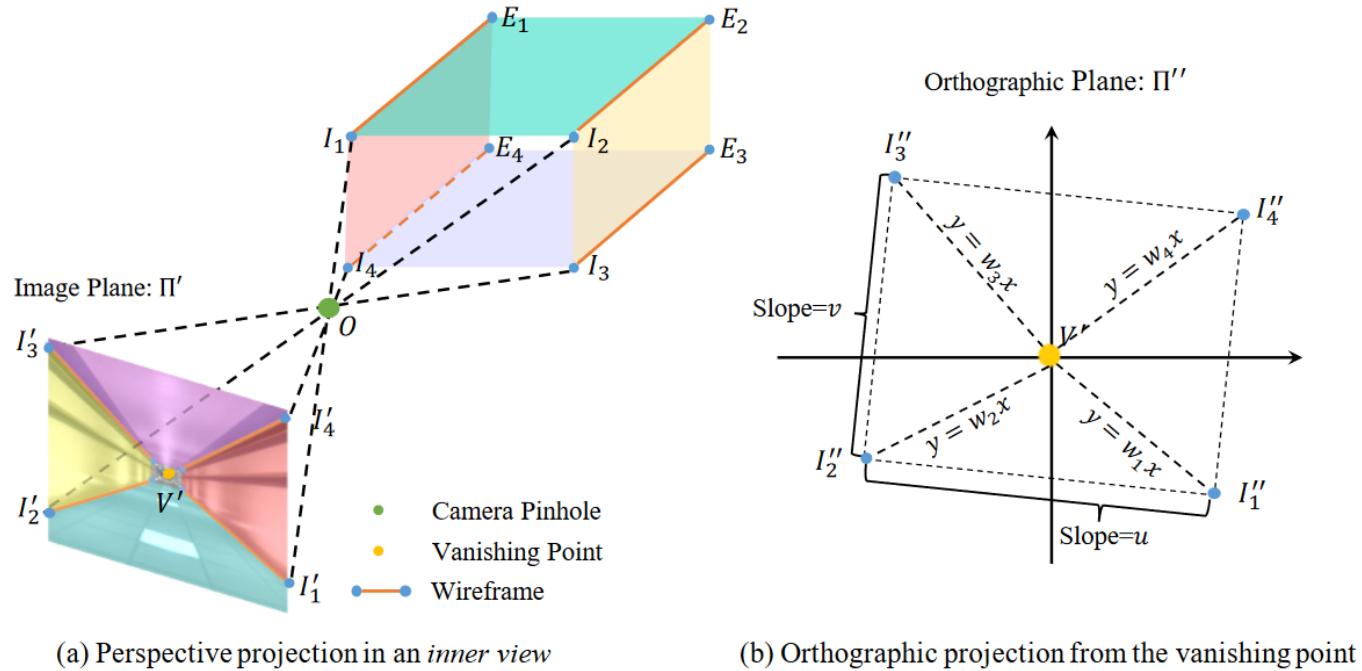


Figure 3: Our Box Program Induction finds the best-fit program that describes the input image (a). It first detects the vanishing point and wireframe segments (b), followed by a filtering step (c). It then constructs a set of candidate plane segmentation maps (d). Given each plane segmentation, it rectifies each plane and infers its regularity structure (e). We rank all candidate box programs by their fitness (f); the starred candidate is the best.

Step 3 - Plane Rectification



(a) Perspective projection in an *inner view*

(b) Orthographic projection from the vanishing point

Figure 10: Illustration of (a) the perspective projection in an inner view of the box (image upside down to be consistent with the projection), and (b) the orthographic projection centered at the vanishing point.

equation system:

$$\begin{cases} y_k = w_k x_k; \quad k = 1, 2, 3, 4 & (I''_k \text{ lies on the ray } VI''_k.) \\ y_1 - y_2 = u(x_1 - x_2) & (\text{definition.}) \\ y_2 - y_3 = v(x_2 - x_3) & (\text{definition.}) \\ y_3 - y_4 = u(x_3 - x_4) & (I''_1 I''_2 \text{ is parallel to } I''_3 I''_4.) \\ y_4 - y_1 = v(x_4 - x_1) & (I''_2 I''_3 \text{ is parallel to } I''_4 I''_1.) \\ x_1^2 + y_1^2 = 1 & (\text{camera-to-subject distance assumption.}) \\ uv = -1 & (I''_1 I''_2 \text{ and } I''_2 I''_3 \text{ are perpendicular to each other.}) \end{cases}$$

Step 4 - Fitness Ranking

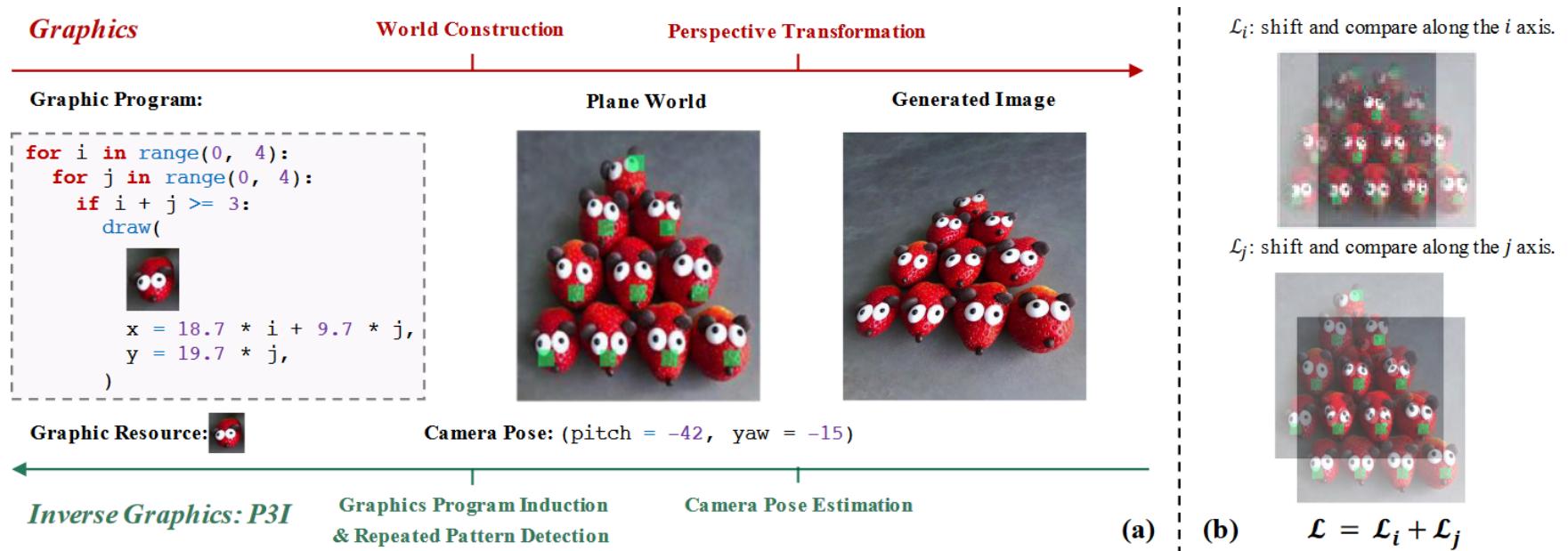


Figure 2: (a) Our model P3I solves an *inverse graphics* problem. Given an input image, P3I jointly infers the camera pose, object locations, and the global scene regularity, which is an inversion of a simplified graphics pipeline. (b) We compute the **fitness** of a program on the image based on a shift-and-compare routine, which we illustrate on a lattice pattern case.

Evaluation

Dataset

- 44 Corridor images, 42 Buildings.
- Collected by Google Images
- Handmade mask for end of the corridor
- Handmade mask for building.

Tasks

- Plane Segmentation
- Image Inpainting
- Image Extrapolation
- View Synthesis

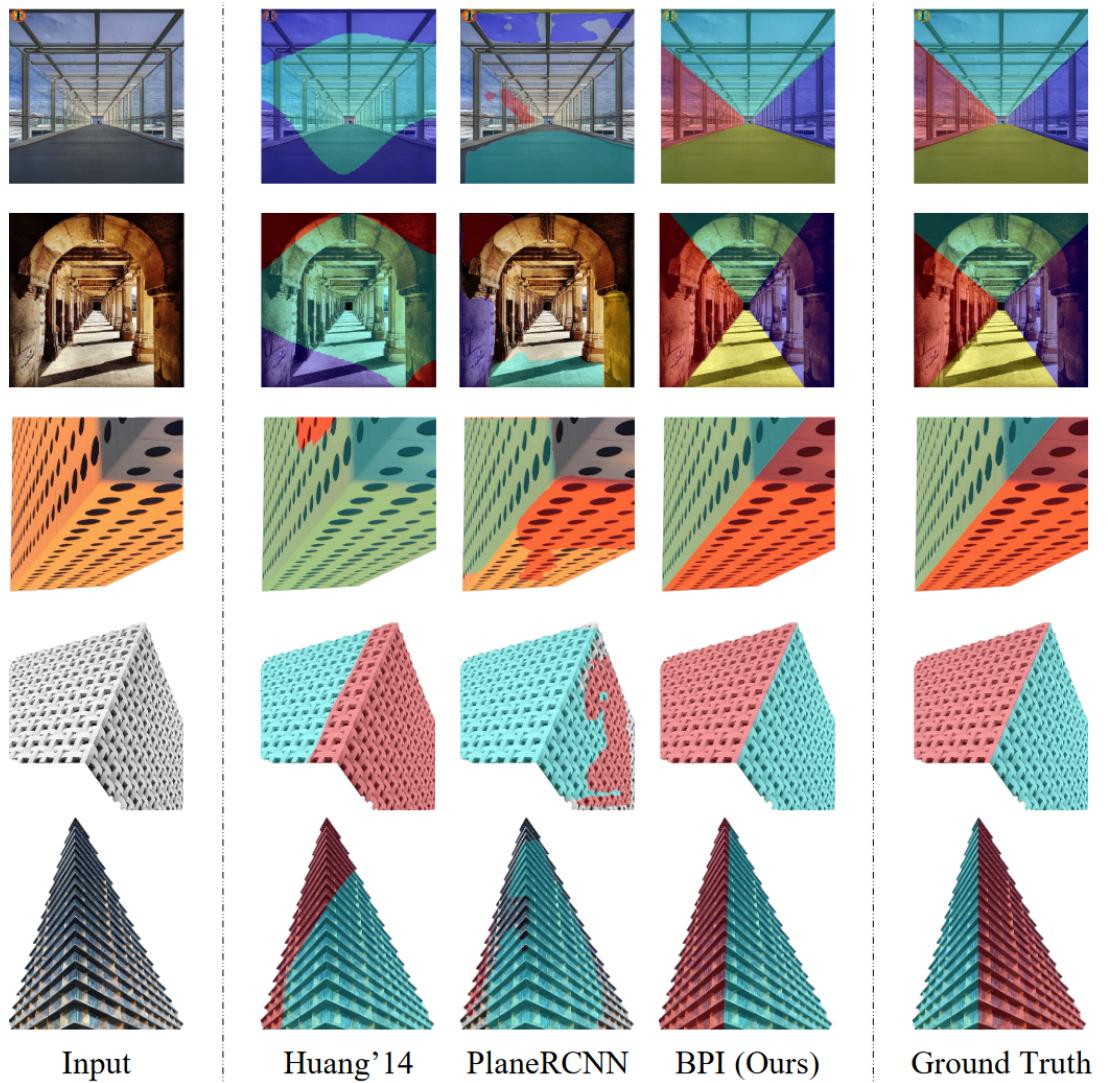


Figure 12: Visualization of the plane segmentation by different methods.

Plane Segmentation (Quantitative Evaluation)

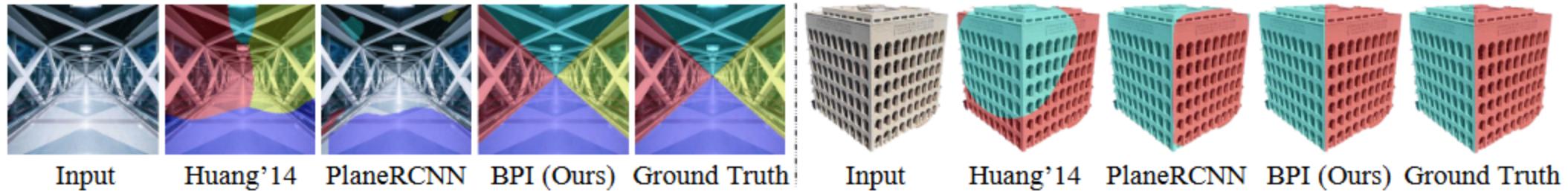


Figure 5: Visualization of the plane segmentation by different methods.

Method	CrdO	CrdC	BldO	BldC
Huang et al. [16]	0.30	0.30	0.73	0.69
PlaneRCNN	0.52	0.52	0.63	0.63
BPI (Ours)	0.84	0.84	0.86	0.86

Table 2: Plane segmentation results in IoU between detected and groundtruth planes. BPI outperforms both baselines on both original (CrdO, BldO) and corrupted images (CrdC, BldC).

Image Inpainting (Quantitative Evaluation)

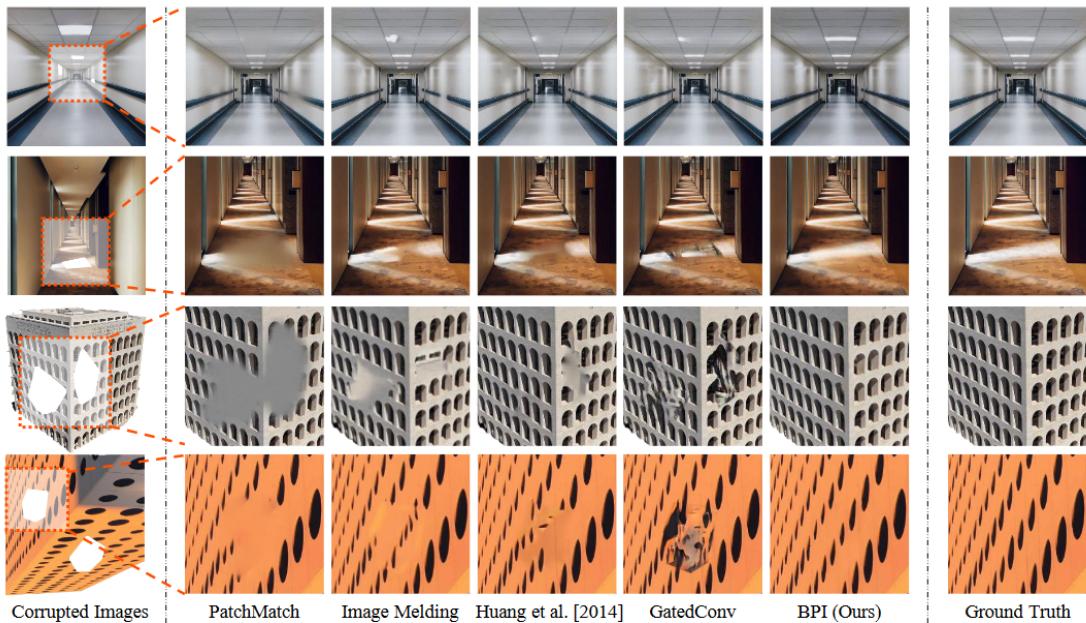


Figure 6: Qualitative results on the task of image inpainting. Compared with the baselines, our model can better preserve the regular structures even if they are sparse or have strong perspective effects.

Method	Corridors				Buildings			
	\mathcal{L}_1 Mean ↓	PSNR ↑	SSIM ↑	LPIPS ↓	\mathcal{L}_1 Mean ↓	PSNR ↑	SSIM ↑	LPIPS ↓
PatchMatch	53.12	31.55	0.9872	0.0215	34.88	32.99	0.9896	0.0072
Image Melding	58.50	30.54	0.9880	0.0174	49.10	30.42	0.9890	0.0134
Huang et al. [16]	51.88	31.41	0.9869	0.0177	26.10	34.87	0.9917	0.0049
GatedConv	44.98	32.32	0.9883	0.0153	55.31	29.54	0.9866	0.0112
BPI (Ours)	38.45	34.21	0.9892	0.0170	26.43	34.86	0.9913	0.0054

Table 3: We compare BPI-guided PatchMatch with both patch-based and learning-based methods on the task of image inpainting. ↑ indicates that the higher the number, the better. **Bold** indicates models that are indistinguishable with the best one under that metric with a linear mixed model. See text for details.

Image Extrapolation (Qualitative Evaluation)

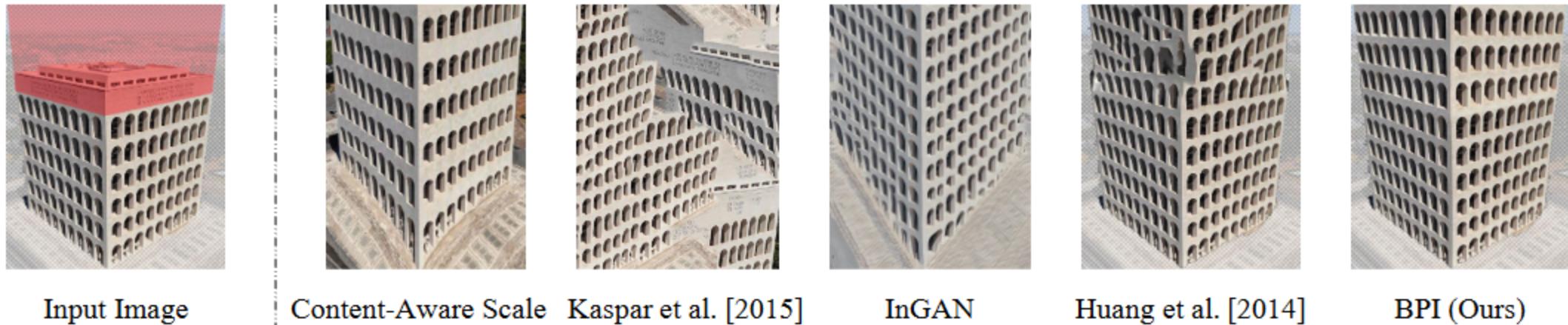


Figure 7: The “extrapolated” buildings. Content-Aware Scale, Kaspar et al. [18], and InGAN fail to preserve the building structure while extrapolating the image: they either generate irregular patterns or change the shape of the planes. Huang et al. [16] fails to preserve the regular structure when inpainting large areas.

Randomly select 15 images and ask 15 participants to rank outputs.

- 61% BPI (This Paper)
- 16% Context-Aware Scaling
- 2% Kaskar et al.
- 16% InGAN
- 5% Huang'14

View Synthesis (Qualitative Evaluation)

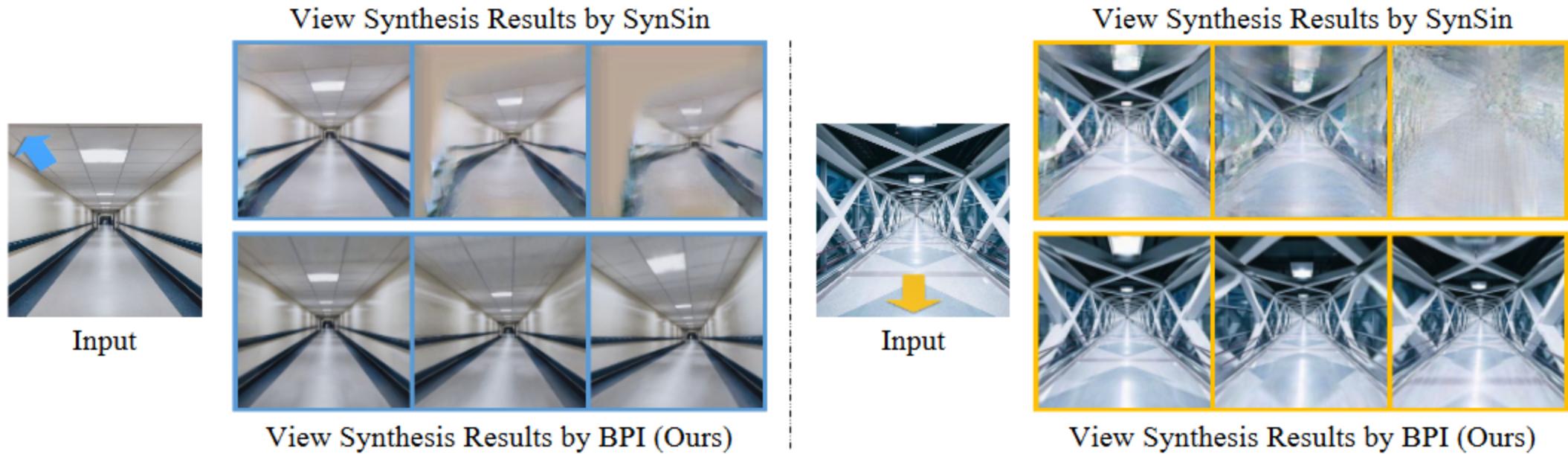


Figure 8: View synthesis from a single image of a corridor. Compared with the learning-based method SynSin, our model better preserves the regular patterns on the walls and also has remarkably fewer artifacts.

Randomly select 20 images from 3 trajectories and ask 10 participants to rank outputs.

- 100% prefer BPI on Trajectory 1
- 94% prefer BPI on Trajectory 2
- 99.5% prefer BPI on Trajectory 3

Failure Cases

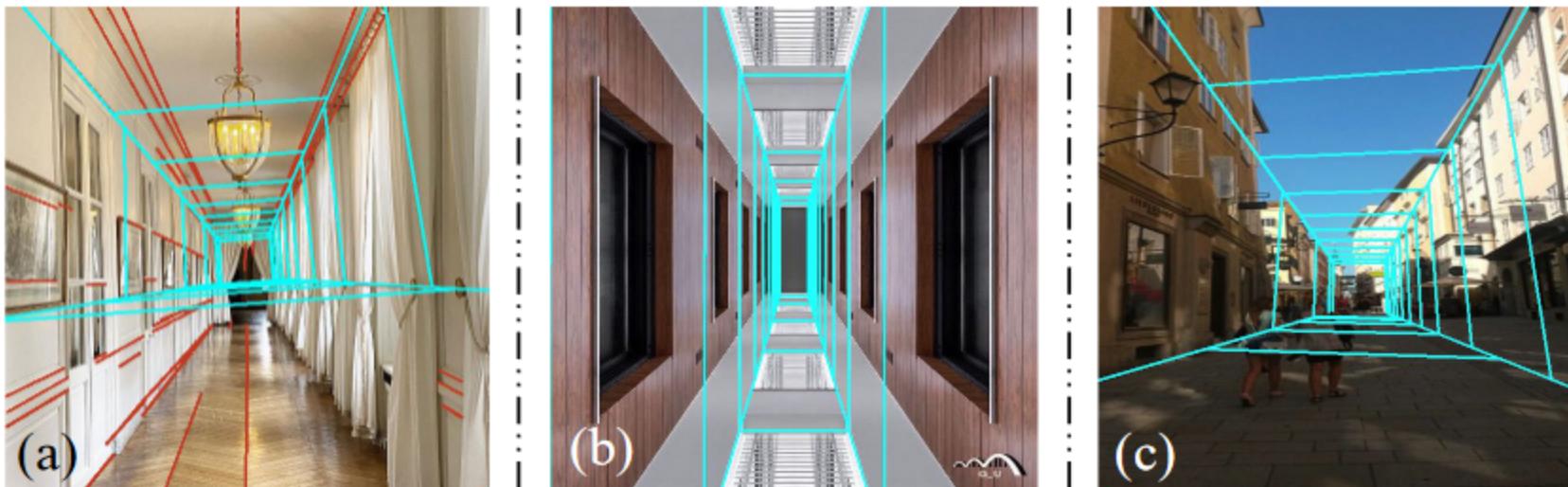


Figure 9: Failure cases. Our model may fail when neural networks misdetect visual cues (a). When image contains a solid color plane, our model may segment the pure white part to other planes (b). The inference might fail on irregular scenes, but could be mitigated with user interaction (c).

Interesting papers

- Perspective plane program induction framework
- The unreasonable Effectiveness of Deep networks as a Perceptual Metric.