A Implementation Details

Base regularization. For the Base region, we render a single downward-facing view $\mathcal{V}_{\text{BASE}} = (I, D, M)$ and apply geometric regularization to ensure a planar floor surface. We minimize the depth variance within the generated floor region to enforce planarity:

$$\mathcal{L}_{\text{flat}} = \text{Var}\left(\{D(u,v) \mid (u,v) \in M\}\right) \tag{10}$$

where M denotes the generation mask and $D(u,v)$ is the rendered depth at pixel (u,v). We additionally constrain surface normals to align with the expected upward floor orientation:

$$\mathcal{L}_{\text{norm}} = 1 - \frac{1}{|M|} \sum_{(u,v)\in M} \left(\mathbf{n}(u,v) \cdot \mathbf{n}_{\text{up}}\right) \tag{11}$$

where $\mathbf{n}(u,v)$ is the rendered surface normal and $\mathbf{n}_{\text{up}} = [0,1,0]^\top$. The combined loss $\mathcal{L}_{\text{BASE}} = \mathcal{L}_{\text{flat}} + \lambda \mathcal{L}_{\text{norm}}$ encourages uniform depth and upward-facing normals, yielding a flat floor that connects the Left and Right regions.

(a) Inpainted BASE Keyframe (b) w/o Regularization (c) w/ Regularization

Figure 10: Base region reconstruction with and without geometric regularization.

Prompts Used. We use Google Vertex AI Imagen 2 [28] for keyframe inpainting and DeeVid AI Video Generation V2.1 [10] for intermediate frame densification.

For the Base region: *"Smooth floor transition strips and threshold for doorway between two adjoining rooms".*

For transition structures:

- **Wing Wall:** *"Thick, protruding interior wall structure."*
- **Pony Wall:** *"Thick, half-height interior partition wall."*
- **Internal Window:** *"Thick interior partition with a central rectangular window."*

We append the following to all structure prompts: *"Filling the masked area, connecting spaces with visible architectural depth from floor to ceiling, seamlessly integrated into both walls. Simple and clean design, full frame."*

For video generation: *"Camera performs a smooth lateral arc around a protruding wall corner at fixed radius, panning between rooms while maintaining constant distance from surfaces and consistent geometry."*

Progressive Anchoring Parameters. We set the maximum number of video generation iterations to 12, extracting 30 frames per video (split for forward/backward passes) and 8 frames for final closing. For pose estimation quality control, we require a minimum of 80 PnP inliers with maximum 1.0 px reprojection error. We employ an adaptive inlier threshold: the first successfully localized frame establishes a baseline, and subsequent frames must maintain at least 55% of this baseline to prevent drift from unreliable estimates. Forward and backward passes converge when reaching within $\tau_d = 0.56$ m of the center line (see Table 3). For feature matching, we use SIFT with brute-force matching (Lowe's ratio 0.75) and PnP-RANSAC (10,000 iterations, 2.0 px threshold, 0.999 confidence). To ensure balanced localization, we require at least 20% of inliers from each anchor space, preventing bias toward a single side.

Distance Threshold Ablation. We ablate the convergence threshold τ_d on five scene pairs (Table 3). A narrower threshold requires the forward and backward passes to approach each other more closely before closing, which increases both the number of localized frames and the number of video generation iterations. At $\tau_d = 0.56$ m the method attains the best mean SSIM while trailing the best PSNR by 0.25 dB and the best LPIPS by 0.0012, at moderate cost. We therefore adopt it as the default.

Table 3: Distance-threshold ablation. † denotes the default setting; bold marks the best value per column. Metrics are computed on a five scene-pair subset, so absolute values differ from the main paper.

τ_d (m)	PSNR↑	SSIM↑	LPIPS↓	#img	#iters
0.80	**27.78**	0.9762	0.0346	32.0	5.4
0.56†	27.53	**0.9780**	0.0303	48.8	7.4
0.20	26.87	0.9763	**0.0291**	71.4	8.8

B Localization Robustness

The quality gate described in Section A determines whether a localized frame is accepted. To characterize its behavior under imperfect inputs, we apply controlled degradations to the anchor RGB and depth as proxies for keyframe-generation and depth-estimation errors, and measure the resulting shift in the recovered camera position relative to the pose obtained from the undegraded input (Table 4).

Pose estimates remain accurate under metric depth scale error, which our scale–shift alignment (Eq. 4) absorbs entirely, and degrade only mildly under per-pixel depth noise, low-frequency depth tilt, and moderate RGB corruption. Under severe detail loss the behavior is qualitatively different: heavy blur and aggressive

Table 4: Localization robustness under controlled degradations. Pose shift is the displacement of the recovered camera position relative to the undegraded input; Reject is the fraction of frames rejected by the quality gate. Arrows indicate values at the weakest and strongest degradation level in each range.

Degradation	Level	Pose shift (mm)	Reject (%)
Depth			
metric scale	±5–20%	0.0	0.0
per-pixel noise	2–20% med. depth	0.4 → 2.4	≈ 0
low-freq. tilt	5–40% med. depth	0.5 → 2.1	≈ 0
RGB			
Gaussian noise	σ/255: 0.02–0.20	4.3 → 8.5	1 → 37
blur	σ: 1–8 px	4.7 → 41	6 → 100
downsample	×2–×8	4.9 → 15	6 → 100

downsampling remove the keypoints the gate relies on, so frames are rejected rather than localized with an incorrect pose. The system therefore fails safe, terminating a pass instead of propagating silent drift into subsequent anchors.

C Runtime Analysis

Table 5 breaks down the processing time for a single transition subregion, corresponding to the 64 minutes reported in Section 7. The iterative loop accounts for 65.6% of the total, and video generation alone for 28.7%. Localization, the stage introduced by Progressive Anchoring, accounts for 7.2%: the cost is dominated by the off-the-shelf video model and by the final NVS reconstruction rather than by the anchoring mechanism itself. Runtime therefore scales primarily with the number of iterations, and can be reduced by adaptive frame sampling and early-convergence detection, provided the localizer is strong enough that early stopping does not trigger prematurely.

Table 5: Runtime breakdown for one transition sub-region on a single RTX 4500 GPU, averaged over 24 evaluated regions. The iterative loop ran for 7.67 iterations on average; indented rows are its constituent stages.

Stage	Time (s)	(min)	Share
Iterative loop (7.67 iters)	2518.9	42.0	65.6%
Video generation	1100.4	18.3	28.7%
Preprocessing	635.1	10.6	16.5%
Volumetric blending	507.3	8.5	13.2%
Localization	276.1	4.6	7.2%
NVS reconstruction	1320.0	22.0	34.4%
Total	3838.9	64.0	100%

D Additional Baseline Details

Camera-Conditioned Generation. Both Gen3C and SVC generate 121 frames along the prescribed trajectory, matching Gen3C's native batch size for fair comparison between generation baselines. Gen3C receives rendered RGB, metric depth, and valid-pixel masks from the voxel model, with 2 key frames (front and rear) packaged as multiview conditioning. SVC receives only 2 conditioning images (front and rear), using trajectory prior and interpolation-based chunking. The 119 generated frames are located between the original conditioning frames.

Camera Pose Estimation. All pose estimation methods receive uniformly-spaced frames from a single generated video, matching our method's total frame count, approximately 50–80 per side (Left, Right). COLMAP uses pycolmap with sequential matching (10-frame overlap) and fixed PINHOLE intrinsics. DUSt3R uses the pre-trained ViT-Large model with consecutive, skip-1, and skip-2 pairs, followed by PointCloudOptimizer alignment (300 iterations, lr=0.01, cosine schedule). MASt3R uses MASt3R_ViTLarge with metric depth and a sliding-window scene graph (window size 5) for scalability, with two-stage sparse global alignment (300+300 iterations, lr=0.07/0.014, joint depth optimization). VGGT uses the 1B-parameter feed-forward model in a single forward pass, subsampled to 35 frames due to quadratic memory scaling of global cross-frame attention. All pose estimation methods are aligned to ground truth via SVD-based rigid alignment with scale computed from the start-to-end distance ratio.

E User Evaluation Details

We recruited 23 participants to evaluate 9 scene pairs with 3 methods: Text2Room [15], NeRFiller [36], and ours. Participants rated each scene on a scale from 1 (poor) to 5 (excellent):

- **Overall Realism (OR):** How realistic does this scene appear?
- **3D Structure Completeness (3DSC):** How complete are the generated 3D structures?
- **Prompt Alignment (PA):** How well do structures reflect the prompt: *"Protruding wall structure with seamless floor threshold transitioning between two rooms"*?
- **Transition Quality (TQ):** How naturally does the generated region connect the two existing spaces?

To counterbalance ordering effects, we divide participants into three groups. Each group evaluated the methods in a different order: Group A (8 participants) viewed Text2Room → NeRFiller → Ours; Group B (9 participants) viewed Ours → Text2Room → NeRFiller; Group C (6 participants) viewed NeRFiller → Ours → Text2Room. All participants evaluated the 9 scene pairs in the same order, yielding 621 total ratings (23 participants × 9 scenes × 3 methods).