SIGGRAPH 2023
LOS ANGELES+ 6-10 AUG

THE PREMIER CONFERENCE & EXHIBITION ON
COMPUTER GRAPHICS & INTERACTIVE TECHNIQUES

POTENTIALLY VISIBLE HIDDEN-VOLUME RENDERING FOR MULTI-VIEW WARPING

ACM TRANS. GRAPHICS, 42(4), 86

JANGHUN KIM AND SUNGKIL LEE*
SUNGKYUNKWAN UNIVERSITY, SOUTH KOREA

Novel-View Warping (in Real-Time Rendering)

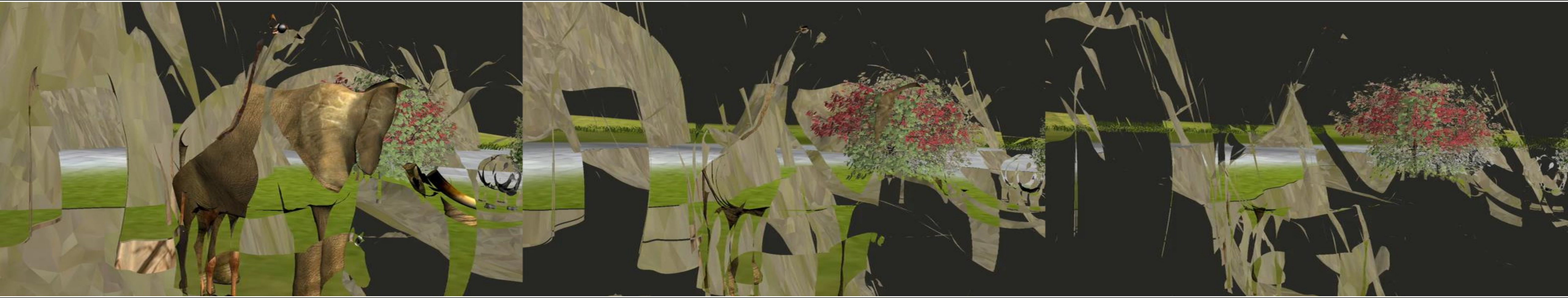
- Single-image warping can introduce disocclusions (or holes), which are not rendered for the known view.

Known view

Novel view with disoccluions

Multi-Fragment Rendering (MFR)

- **A-buffer, k-buffer, and Depth Peeling (DP)**
  - typically used for transparency or global illumination
  - can handle disocclusions by including **hidden fragments** in warping.



Hidden layer 1 **Hidden layer 2** **Hidden layer 3**

* A-buffer [Carpenter 1984; Yang et al. 2010], k-buffer [Bavoil et al. 2007], and Depth Peeling (DP) [Everitt 2001; Mammen 1989]

Redundancy of MFR

- **Many of the fragments**
  - are **invisible** from any novel views and
  - do **not contribute** to the final outcome.

Hidden Layer 1 Hidden Layer 2 Hidden Layer 3

Challenges

- **Reduction of redundancy in MFR and warping**

- **However,**
  - In general, disocclusions are revealed **after warping**.
  - Also, multi-view warping even requires to be **iterated**.

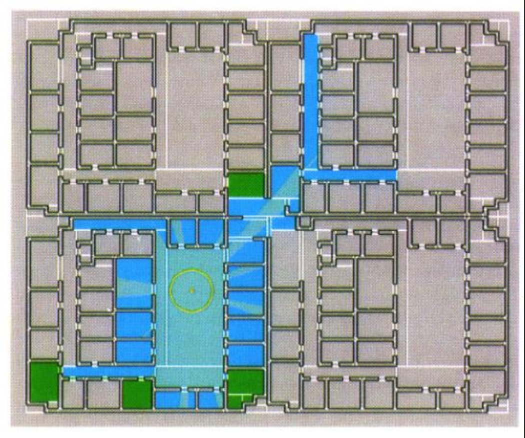
Our Goal

- **Early test of the visibilities for fragment culling in MFR**
  - Capture fragments **for the known views**, but pre-test their visibilities **against novel views**.
  - in particular for Depth Peeling (DP)

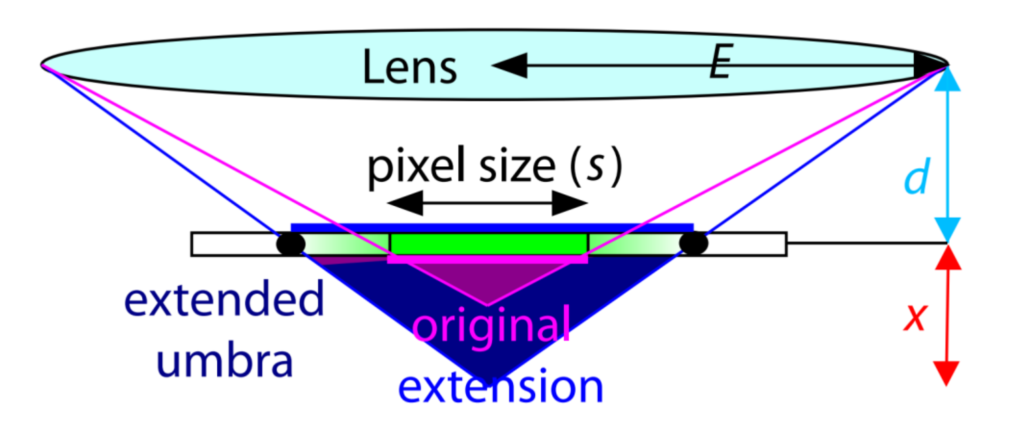
Hidden layer 1 **Hidden layer 2** **Hidden layer 3**

Previous Solutions

- **Potentially Visible Set (PVS)**
  - Offline visibility culling [Teller and Séquin 1991; Cohen-Or et al. 2003]
- **Umbra fragment culling in Depth-of-Field (DOF) rendering**
  - Based on pixel-as-geometry occluders [Lee et al. 2010]



Visibility preprocessing for interactive walkthroughs [Teller and Sequin 1991]



Real-Time Lens Blur Effects and Focus Control [Lee et al. 2010]

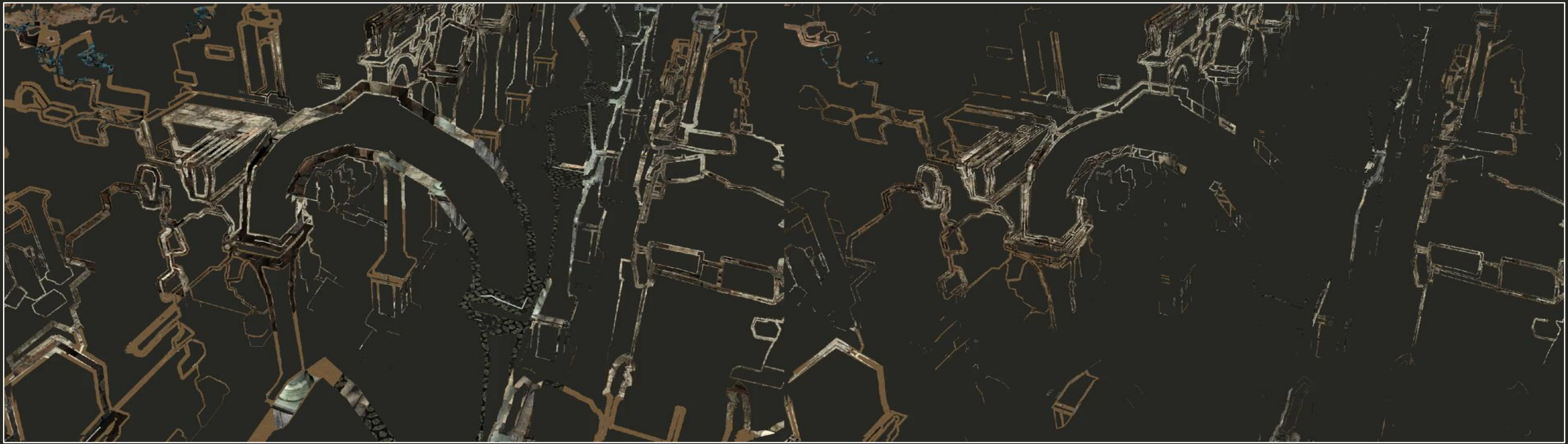
Our Contributions

- **Potentially Visible Hidden Volume (PVHV)**
  - **Definition and modeling of PVHVs** for MFR
  - PVHVs are 3D volumes that are **hidden at the known source view** but **visible at novel views**.

- **Effective Depth Peeling (EDP) Algorithm**
  - PVHV-based **on-the-fly** real-time **fragment culling**

Benefit of Our Solution

- **Multi-view warping with EDP**
  - produce **fewer fragments/layers** for the same quality
  - higher rendering **performance**
  - **higher memory efficiency** (packing from sparser fragments)

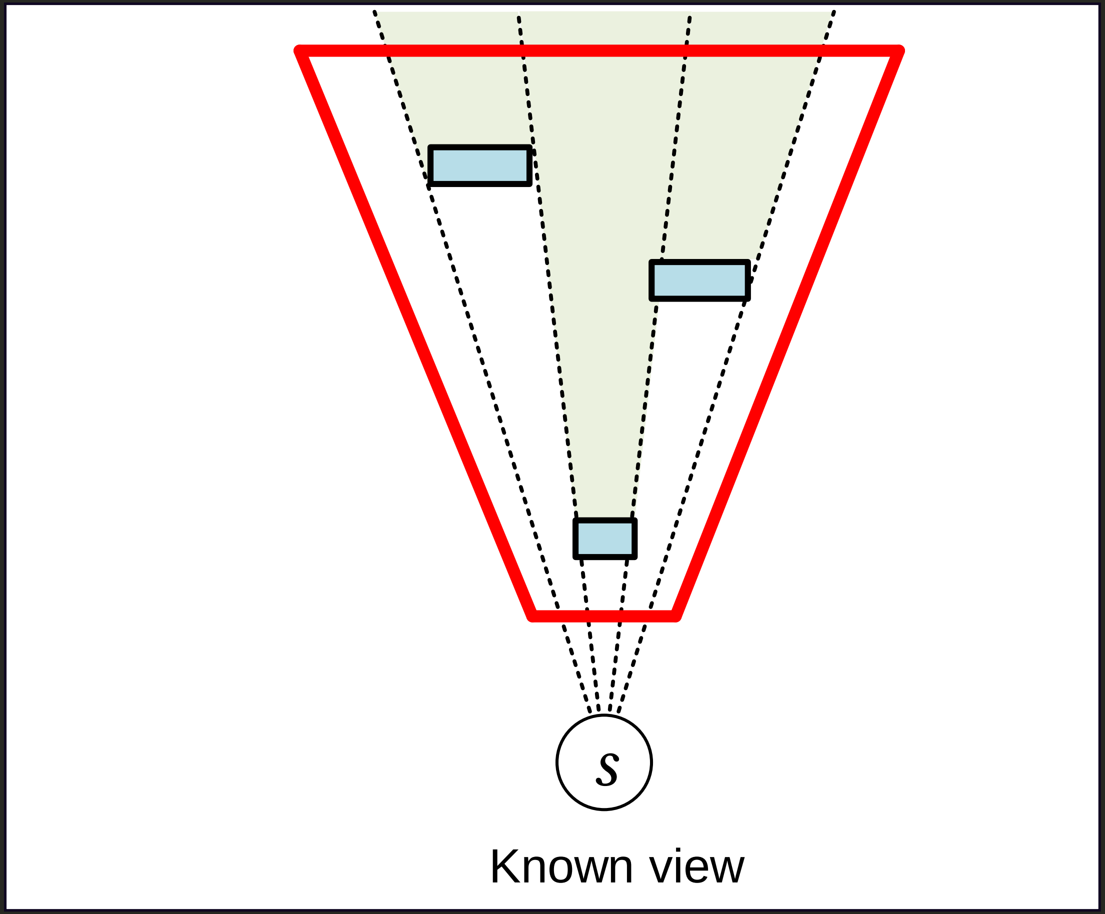


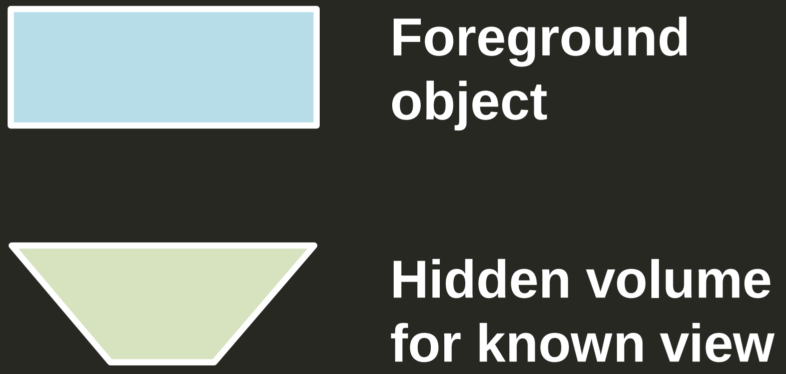
Hidden Layer 1 **Hidden Layer 2**

Potentially Visible Hidden Volumes (PVHVs)

Definition of PVHV:

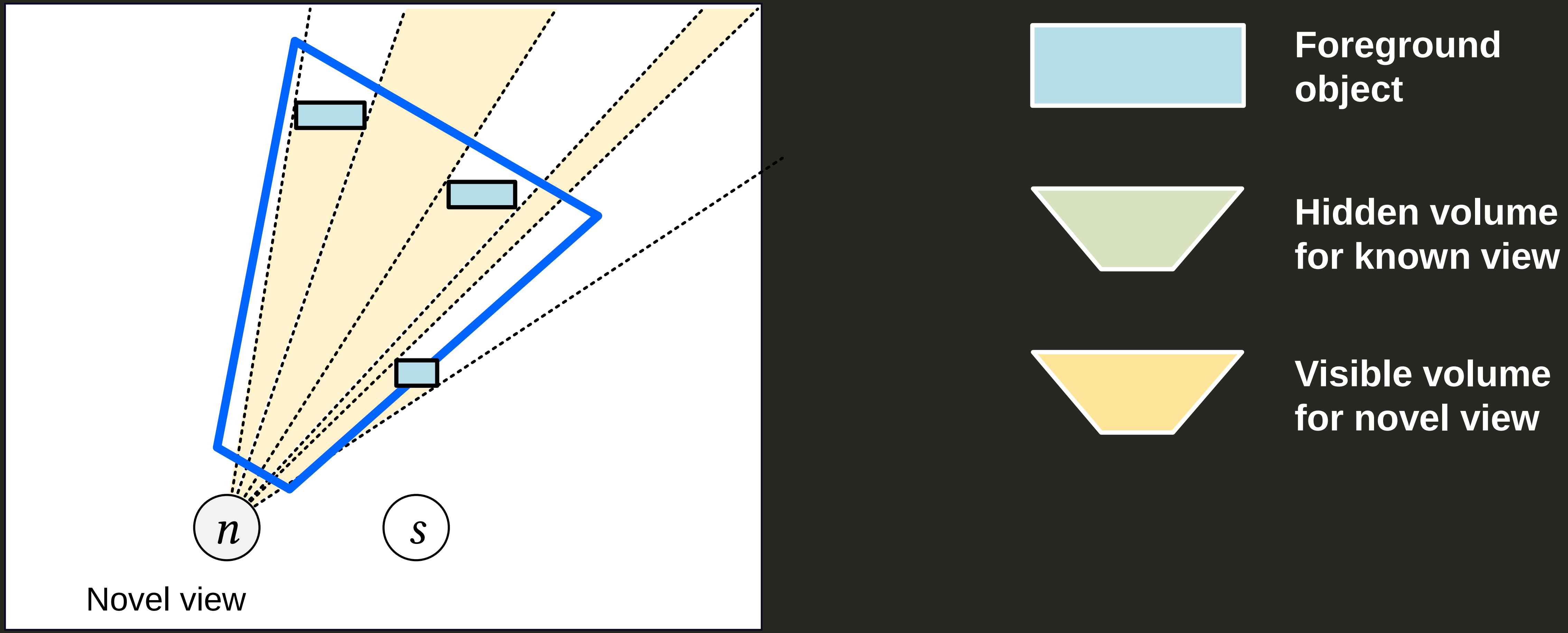
- O_s : Hidden volume from s





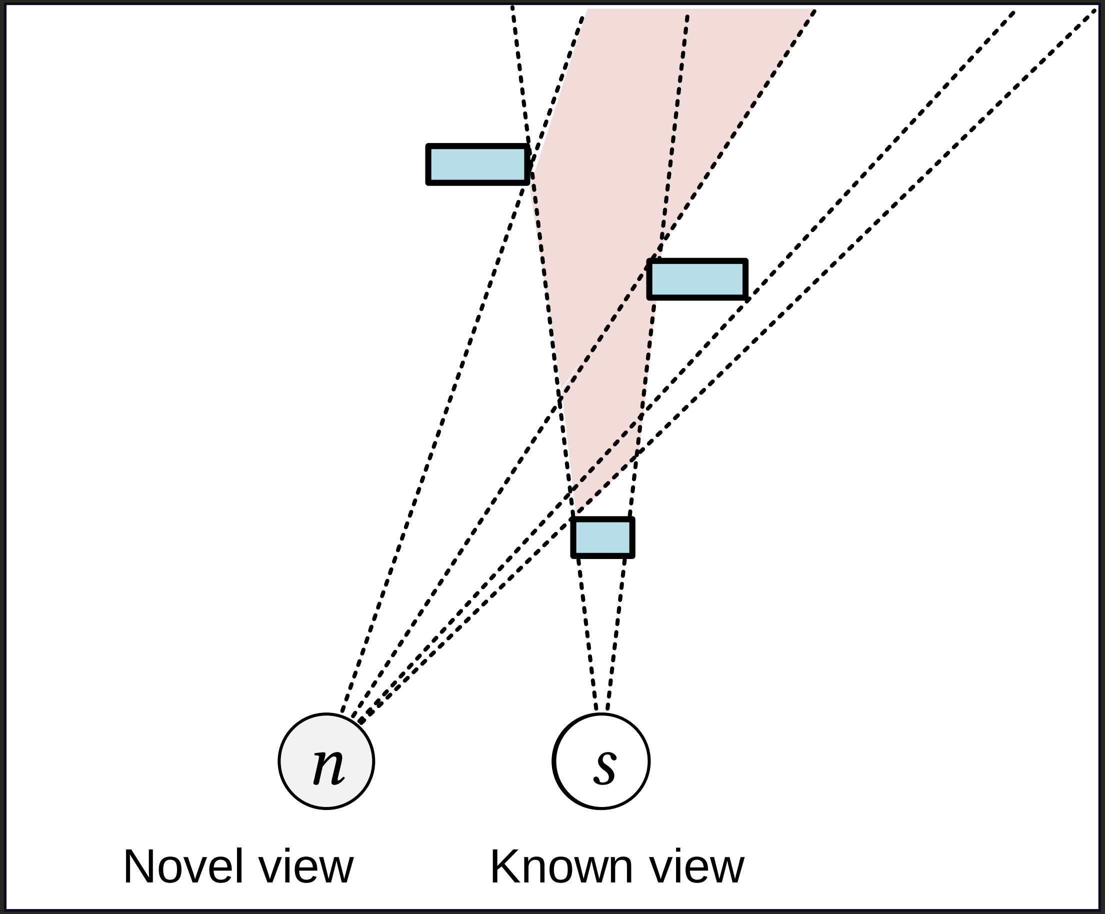
Definition of PVHV:

- V_n : Visible volume from n



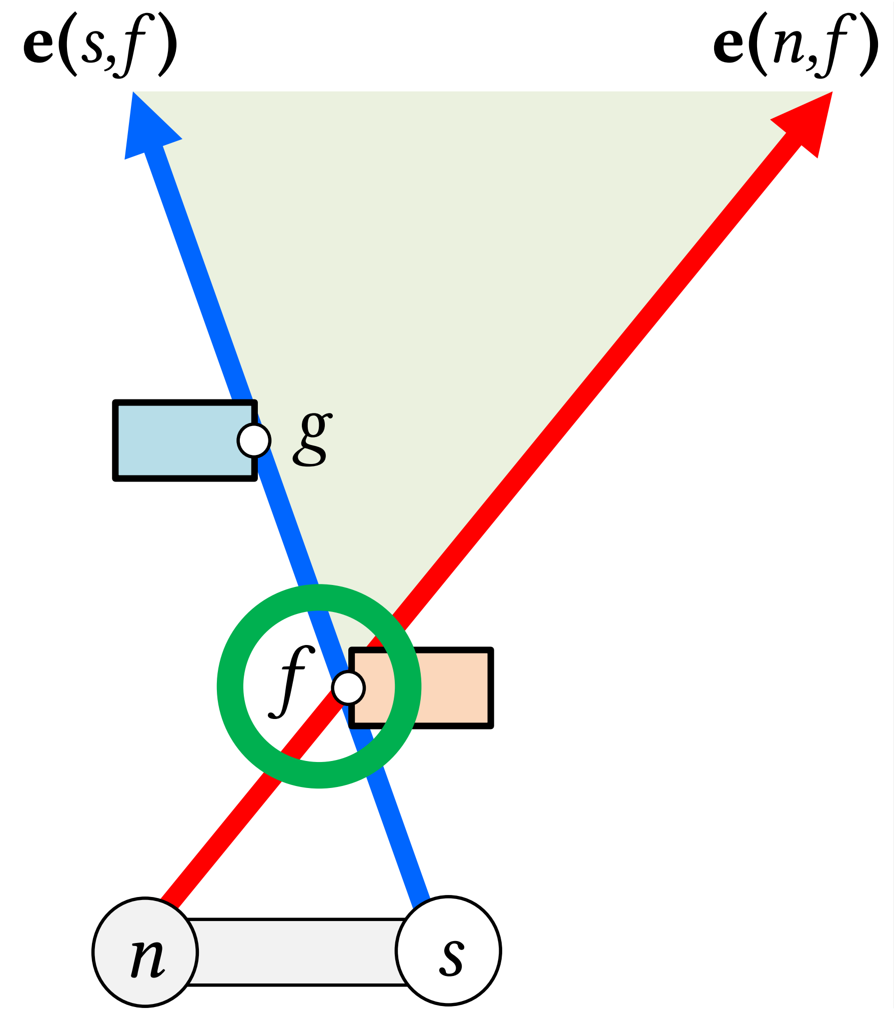
Definition of PVHV:

- **PVHV :** $H_s(n) = O_s \cap V_n$



- **Foreground object**
- **Hidden volume for known view**
- **Visible volume for novel view**
- **Potentially Visible Hidden Volume (PVHV)**

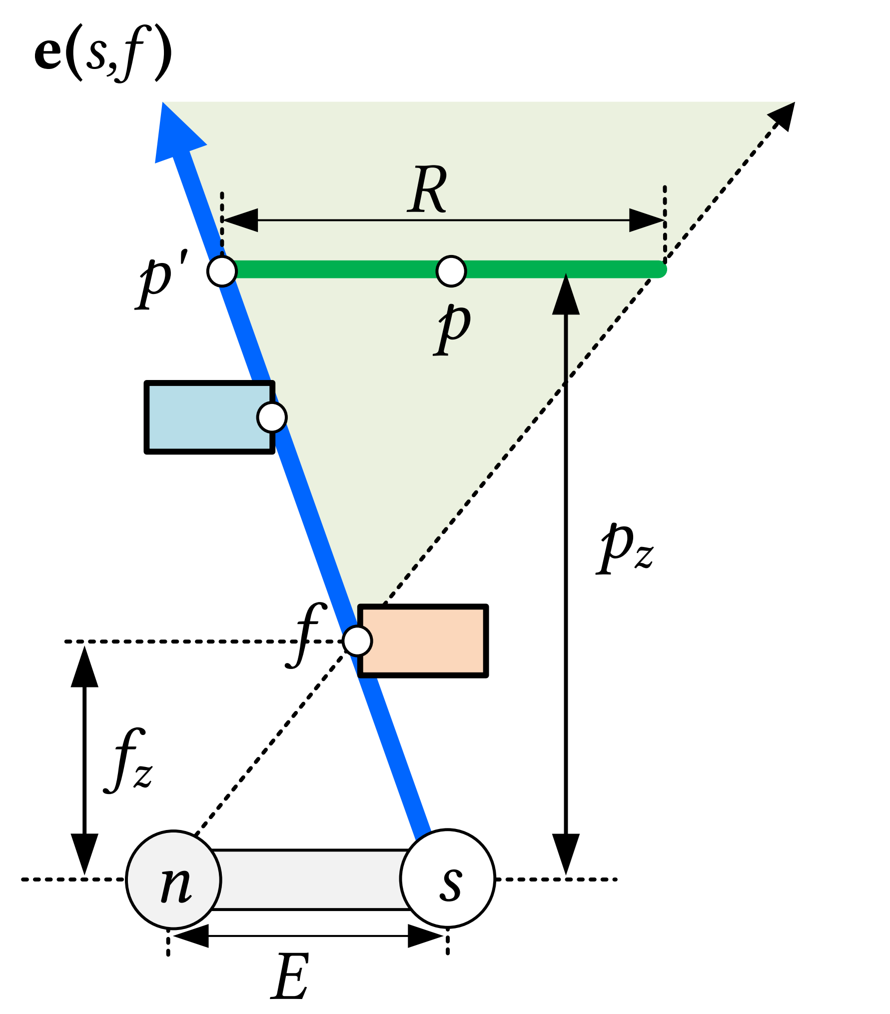
PVHVs for Linear Views



Linear Novel Views

- **The simplest shape of a PVHV**
- **Key elements for finding PVHV**
  - ***inner blocker fragment*** f
  - **two edge rays** passing through f

Local Circle of Confusion (LCOC)



Linear Novel Views

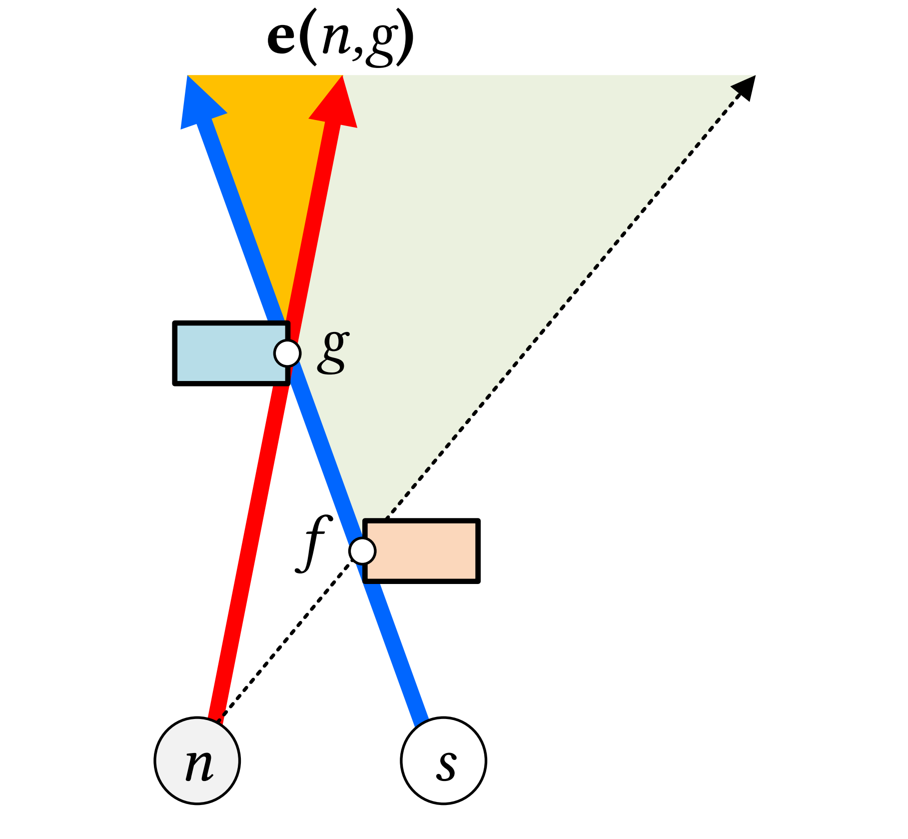
- **PVHV is characterized by LCOC (——)**
  - Similar to COC in DOF rendering
  - LCOC Radius from triangle similarity:

$$R(p, f) = \left(\frac{p_z - f_z}{f_z}\right) E$$

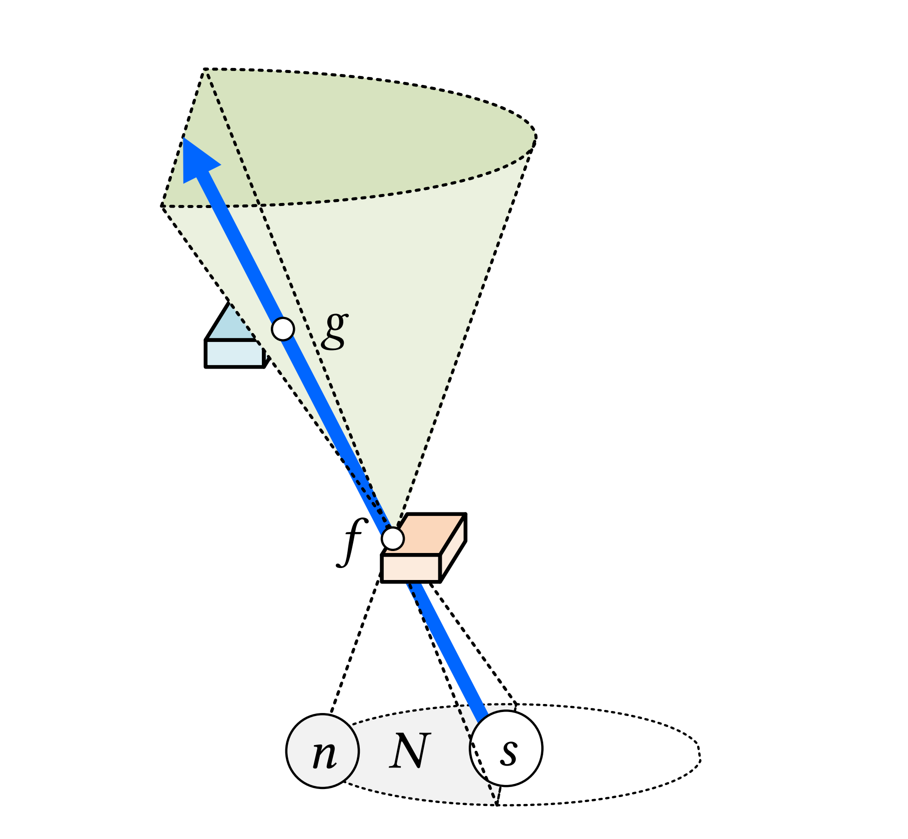
p : Incoming fragment to test
f : Inner blocker fragment
E : distance from s to n

Other Types of PVHV

- It is possible to extend PVHVs to point and areal view types.



Point Novel View



Areal Novel Views

Effective Depth Peeling (EDP)

Depth Peeling

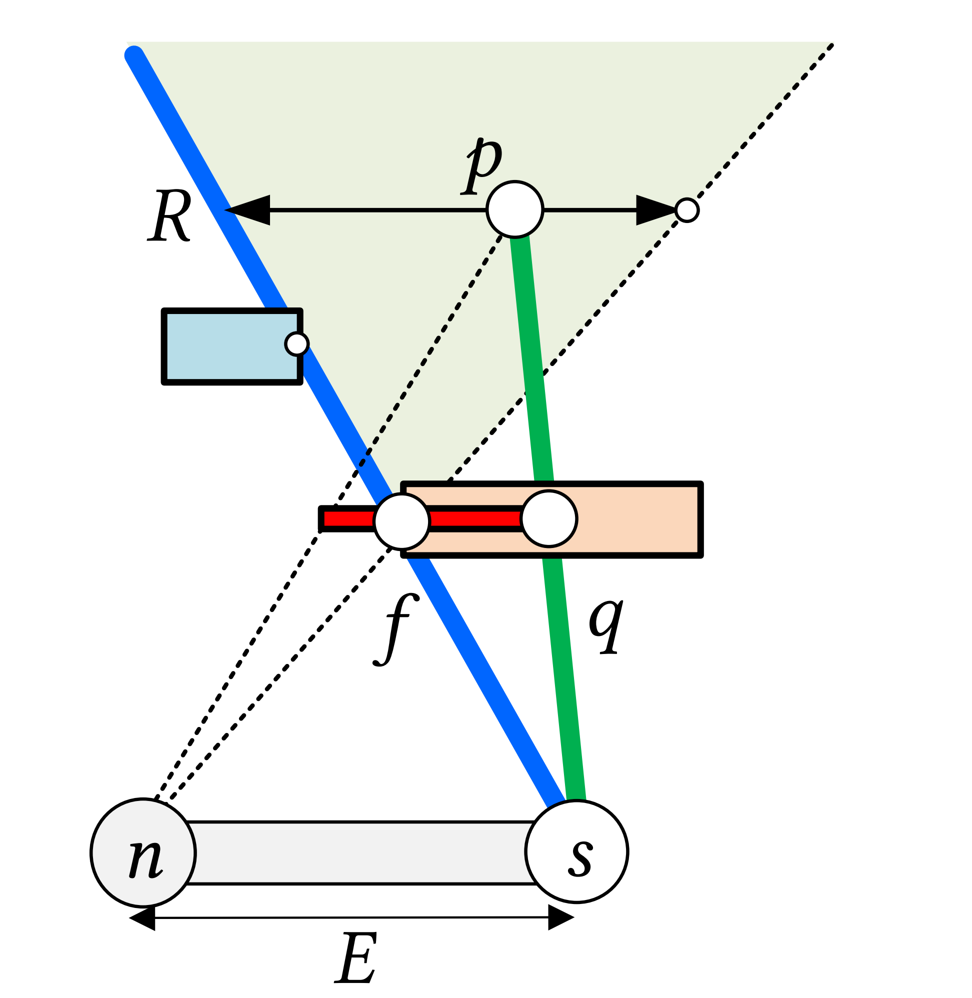
- **Standard Depth Peeling (DP) [Everitt 2001]**
  - captures all hidden fragments

- **Our Effective Depth Peeling (EDP)**
  - DP + PVHV-driven fragment culling

Problem for Efficient Implementation

- **Problem:**
  - PVHVs require to find edges.
  - **Finding edges explicitly** needs to be avoided for efficiency.
  - We just want to test edge exists (rather than where the edges are).

EDP: Backward-Search Algorithm



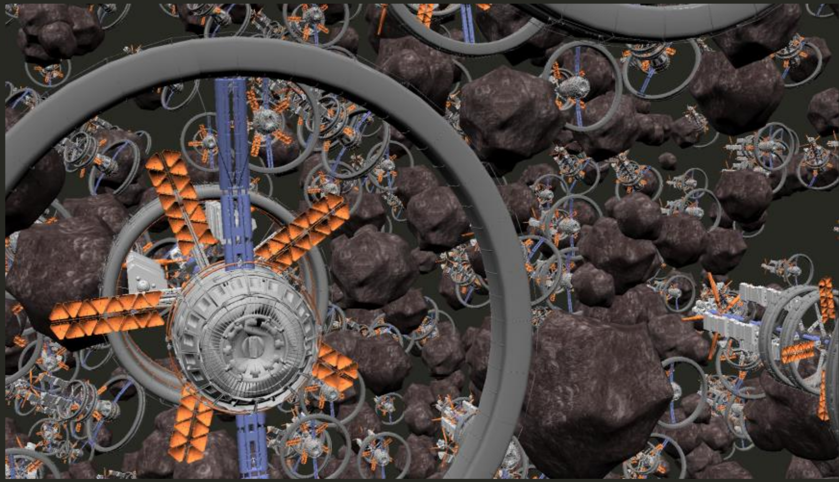
(b) backward search

- **Actually, we need f_z for LCOC.**
  - This needs precise edge detection.
- **Assuming blocker is almost flat**
  - blocker depth: $q_z \cong f_z$
- **Search bound for finding edges**
  - LCOC projection ( ) onto blocker
  - **When an edge exists**, fragment p is **in PVHV**; p has be kept during DP.

Experimental Analysis

Test Configurations

- **Experimental configurations**
  - NVIDIA GTX 3090, Full-HD (1920×1080), OpenGL 4.6
  - Three camera-animated scenes (1.25M—242M faces, 2.2—36K objects)



Fragments in the First Hidden Layer

- **Resulting fragments are already sparse**
  - Most fragments behind large occluders are discarded well

Baseline DP [Everitt 2001]

Fragments: 0.994

Umbra DP [Lee et al. 2010]

Fragments: 0.915

Effective DP [DP+PVHV; ours]

Fragments: 0.392

Performance in Multi-View DOF Warping

- **Speed-ups with respect to Standard DP (1024 views)**
  - Ruins scene: **3.2-3.4**×
  - Safari scene: **2.3-2.9**×
  - Satellites scene: **2.4**×

Memory Efficiency

- **Packed EDP (PEDP)**
  - GPU-based linked list, storing only sparse fragments, can greatly reduce the memory consumption.

Scene	*E* (mm)	Memory (MB)			Memory Efficiency		
		PEDP	EDP	UDP	PEDP	EDP	UDP
RU	50	**17.7**	316.4 (10)	316.4 (10)	**37.5×**	2.1×	2.1×
	100	**26.5**	379.7 (12)	379.7 (12)	**25.1×**	1.8×	1.8×
SF	50	**16.4**	221.5 (07)	316.4 (10)	**61.6×**	4.6×	3.2×
	100	**20.5**	316.4 (10)	379.7 (12)	**49.5×**	3.2×	2.7×
ST	50	**9.1**	94.9 (03)	94.9 (03)	**166.1×**	16.0×	16.0×
	1000	**20.6**	443.0 (14)	759.4 (24)	**73.7×**	3.4×	2.0×

Thank you for attention!