

1 Occlusion Fusion

Our approach can not perform occlusion fusion and this can make the PVS computation more conservative at times. However, such cases only arise if a frustum’s boundary lies along the silhouette edges of one of the objects. Otherwise, the individual objects would eventually become the blockers for the sub-frusta generated by subdivision. This is shown in Figure 1(a), where the silhouette edges of both the objects $V1$ and $V2$ lie inside one of the frustum. Based on our blocker computation algorithm, $V3$ will be in the PVS of $F3$. Notice, even in this case the object $H2$ is not in the PVS of $F3$. Further, depending on the distribution of uniform frusta inside the viewing frustum the object $H1$ will not be visible as $V1$ becomes a blocker for $F4$ and $V2$ is a blocker for $F5$.

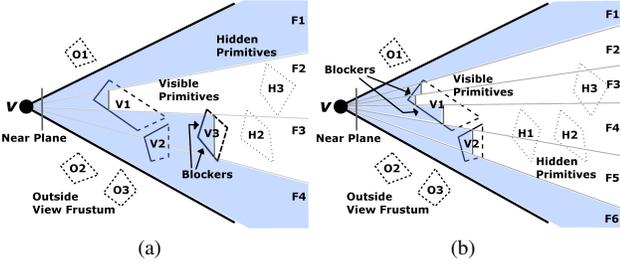


Figure 1: Occlusion Fusion: (a) No occlusion fusion as the frustum lies along the silhouette edges of object $V1$ and $V2$. (b) Occlusion fusion happens implicitly as no frustum aligns along the silhouette edges of multiple objects.

2 Ray Tracing Visible Set Computation

Computing the exact visible set of primitives using sample-based approaches such as ray tracing is rather non-trivial for complex models such as the powerplant. In order to estimate the (object-space) exact visible set, we use ray tracing at an image resolution of 1024×1024 pixels and performed additional supersampling per pixels to generate a higher resolution. We compute all the intersecting triangles (i.e. visible set) with the rays and they are a subset of π_{exact} . Note that the size of the visible set computed by ray tracing is a non-decreasing function as the number of supersamples increase. Moreover, the size of the visible set is also bounded by π_{exact} , the visible set should converge as we increase the number of supersamples.

However, as we increased supersampling (from 2×2 to 32×32), the size of the visible set computed by ray tracing does not converge well on some of the models. Figure 2 shows the convergence behavior on different model. Each curve shows how the visible set size increase with the supersampling.

Ideally, each curve should converge towards a fixed value. However, we don’t observe even at a very high resolution of $32K \times 32K$. This implies that sample-based methods may not be accurate unless the resolution is extremely high.

3 Frustum Tracing PVS Convergence

In this section, we highlight the performance of FastV on different models. Figure 3 shows how the PVS size varies relative to our baseline (4096×4096 uniform frusta) as the number of frusta traced increases. For each model, we consider a path with more than 200 viewpoints and compute the PVS ratio for varying resolutions of the frusta. In each case, we observe that as the frusta size decreases, the

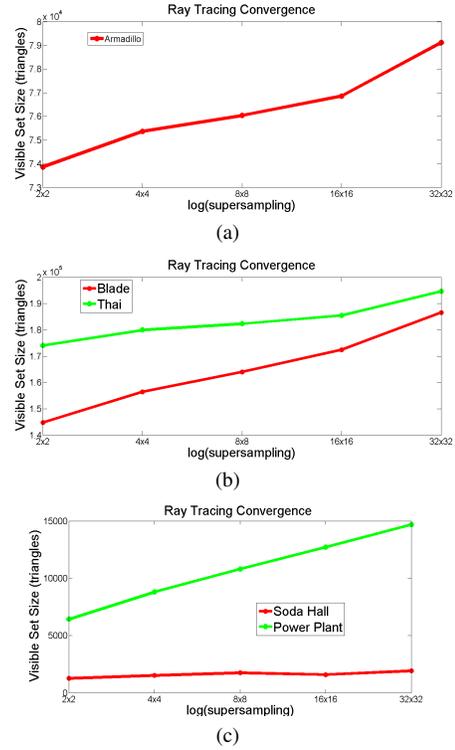


Figure 2: Convergence behavior of the visible set computed by the ray tracer on different models with increasing supersampling.

PVS size converges to the π_{exact} across all the frames for different models. This data indicates that FastV indeed converges towards π_{exact} and is more reliable than sample-based approaches.

4 Comparison with Beam Tracing

In this section we compare PVS computed by FastV using $4K \times 4K$ uniform frusta with a PVS computed by an efficient beam tracer [Overbeck et al. 2007]. We denote the PVS computed by FastV as $PVS_{4K \times 4K}$ and the PVS computed by the beam tracer as PVS_{BT} . We choose 10 key frames from the armadillo sequence in the supplement video and compare $PVS_{4K \times 4K}$ and PVS_{BT} for the camera position of these key frames (see Figure 5). The key frames are equally spaced starting at Frame 1 (the sequence has a total of 270 key frames). Figure 4 and Table 1 summarizes our results. Note that our PVS converges to within 10% of the exact from-point beam tracing solution.

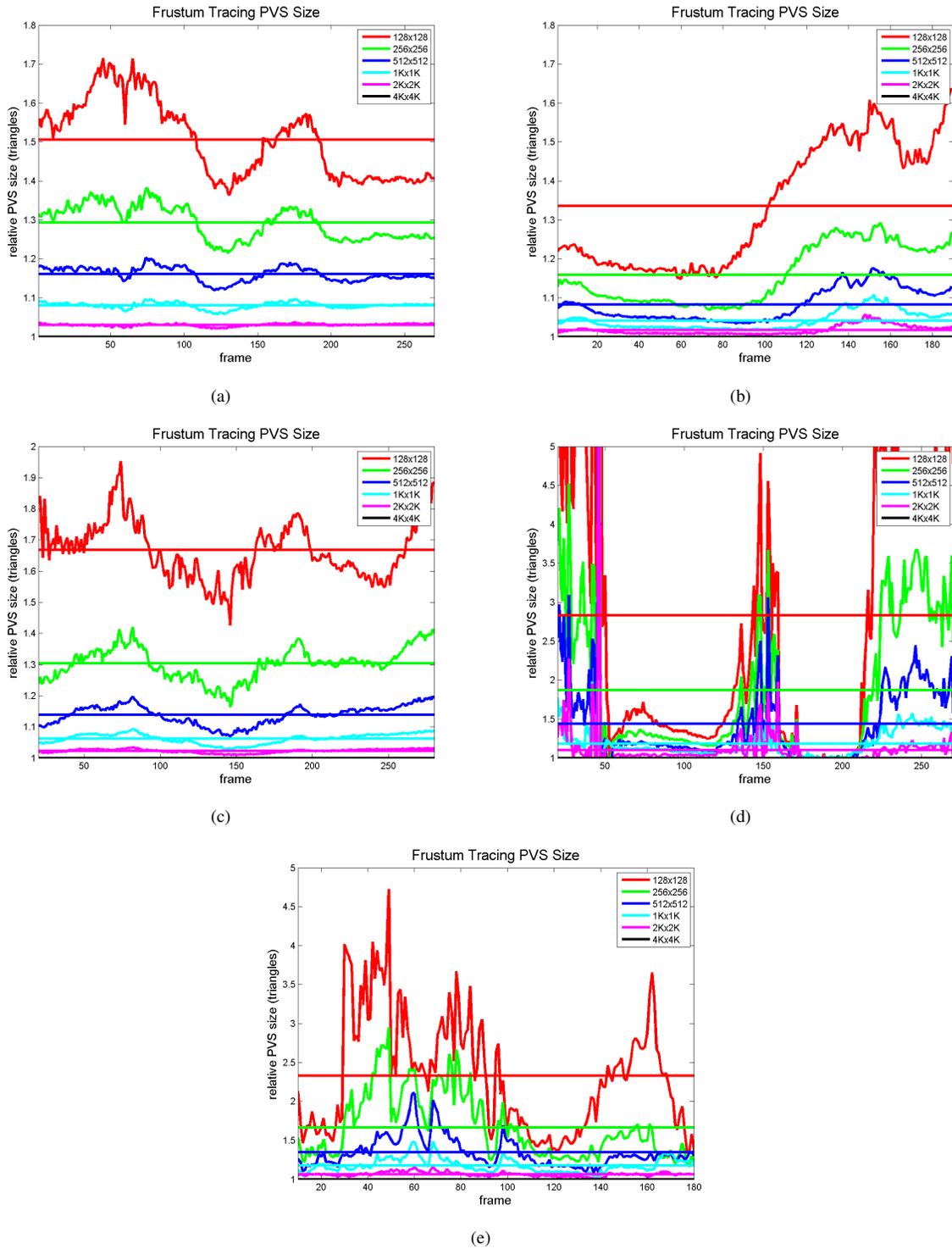


Figure 3: Plots of PVS size as computed by frustum tracing across frames for different models. Solid horizontal lines indicate average PVS size across all frames for the appropriate frustum resolution. The spacing between curves for any given frame in each model reduces to zero with increasing frustum resolution, indicating convergent behavior. (a) armadillo, (b) blade, (c) thai, (d) sodahall, (e) powerplant.

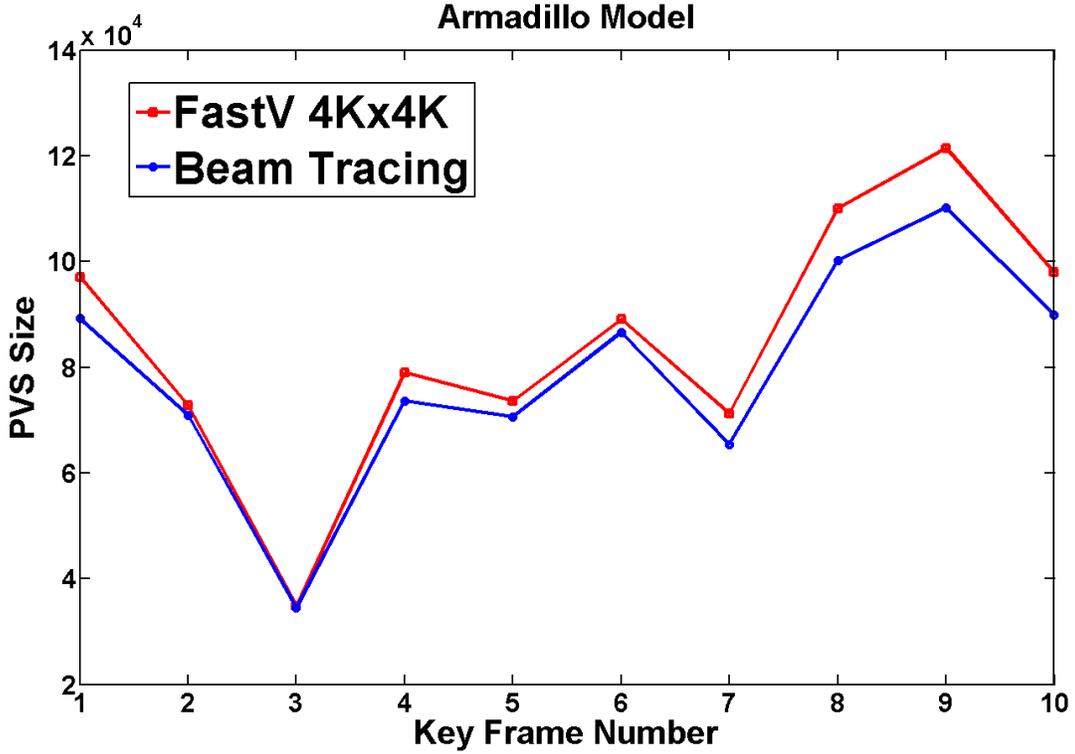


Figure 4: Comparison of PVS sizes of our baseline PVS ($PVS_{4K \times 4K}$) and a beam tracing [Overbeck et al. 2007] PVS (PVS_{BT}).

Frame Number	PVS Size		$\frac{PVS_{4K \times 4K} - PVS_{BT}}{PVS_{BT}} \times 100$
	FastV 4Kx4K $PVS_{4K \times 4K}$	Beam Tracing PVS_{BT}	
1	97079	89192	8.84
2	72758	70948	2.55
3	34877	34434	1.28
4	78958	73660	7.19
5	73558	70612	4.17
6	89067	86596	2.85
7	71253	65361	9.01
8	110062	100248	9.78
9	121428	110210	10.17
10	97950	89846	9.01

Table 1: Difference in the PVS sizes of our baseline PVS ($PVS_{4K \times 4K}$) and a beam tracing [Overbeck et al. 2007] PVS (PVS_{BT}) as a percentage of PVS_{BT} .

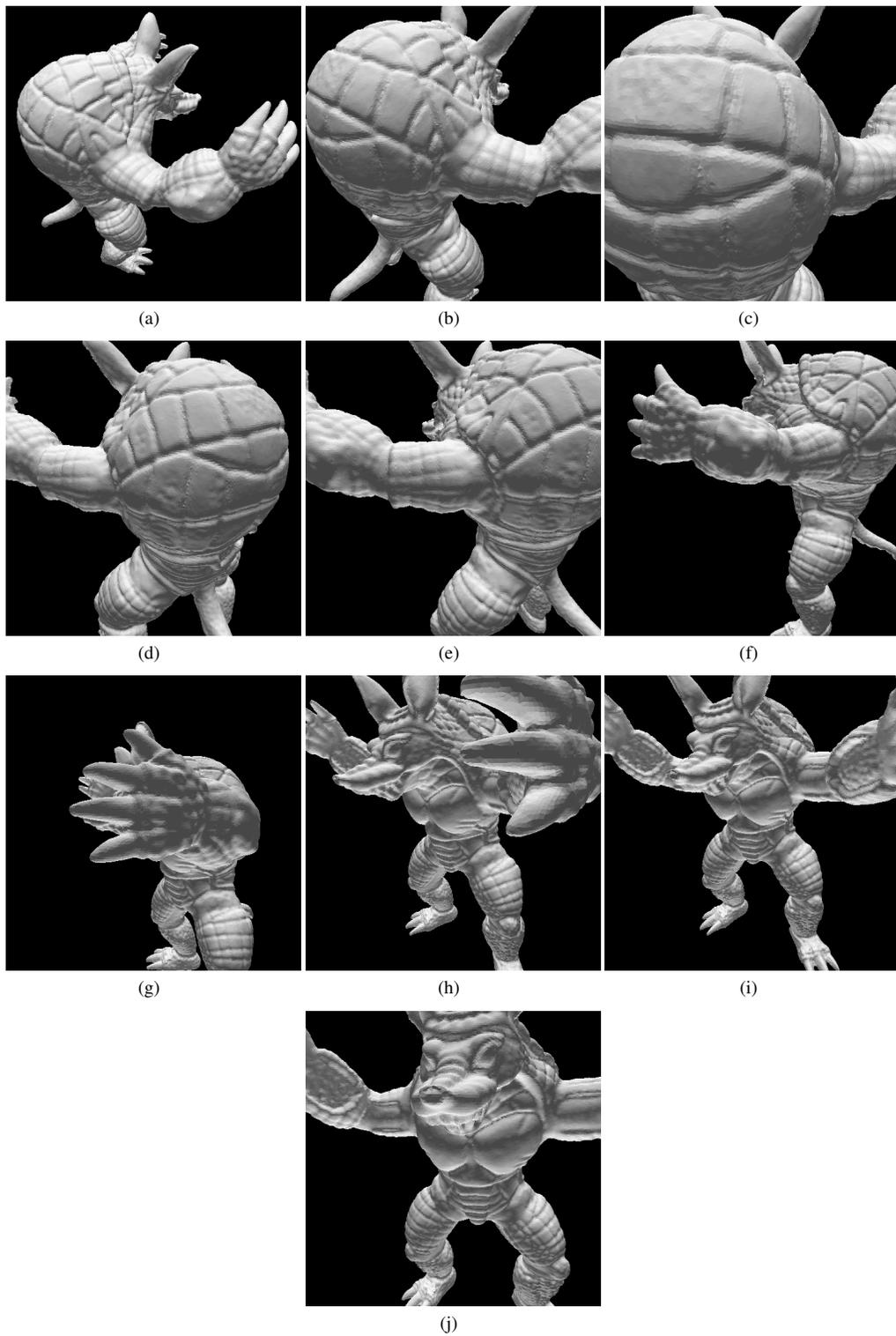


Figure 5: Ten key frames used for comparing the from-point PVS computed by FastV and an efficient beam tracing implementation [Overbeck et al. 2007]. The PVS rendered in the images above was computed using the beam tracer [Overbeck et al. 2007].