Efficient and Robust Radiance Transfer for Probeless Photorealistic Augmented Reality

Lukas Gruber [1]* Tobias Langlotz [1]* Pradeep Sen [2]† Tobias Höllerer [2]‡ Dieter Schmalstieg [1]◇

[1]Graz University of Technology [2]University of California, Santa Barbara

Figure 1: Examples of applying our adaptive radiance transfer computations for probeless light estimation in Augmented Reality. (Left) Augmented Reality based interior shopping application. (Middle, Right) Physical gaming using depth cameras.

ABSTRACT

Photorealistic Augmented Reality (AR) requires knowledge of the scene geometry and environment lighting to compute photometric registration. Recent work has introduced probeless photometric registration, where environment lighting is estimated directly from observations of reflections in the scene rather than through an invasive probe such as a reflective ball. However, computing the dense radiance transfer of a dynamically changing scene is computationally challenging. In this work, we present an improved radiance transfer sampling approach, which combines adaptive sampling in image and visibility space with robust caching of radiance transfer to yield real time framerates for photorealistic AR scenes with dynamically changing scene geometry and environment lighting.

Keywords: Augmented reality, photometric registration, radiance transfer

Index Terms: H.5.1 [Information Interfaces and Presentation]: Artificial, Augmented,Virtual Realities—;I.4.8 [Image Processing and Computer Vision]: Photometric registration—3D Reconstruction;I.3.3 [Computer Graphics]: Image Generation—Ray Tracing-Spherical Harmonics;

1 INTRODUCTION

Realistic light interaction between real and virtual objects is a key factor for photorealistic Augmented Reality (AR). Computing such interaction requires estimation of real world lighting (photometric registration), estimation of scene geometry (3D reconstruction) and finally radiance transfer (RT) computation between virtual and real objects. Since the first two – real world lighting and real world geometry – can change dynamically, and RT depends on both, all three must be computed in real time. Moreover, sensor limitations will necessarily introduce errors and imperfections in the 3D reconstruction and photometric registration. Handling such errors robustly limits the choice of techniques. Consequently, existing approaches have usually concentrated on a certain aspect of the problem and made simplifying assumptions for the remaining aspects.

Concerning geometry, scenes may be assumed to be small and have simple geometry, so the computational effort can be bounded [14, 9, 13, 22]. This can be combined with an assumption that scenes are static, with the exception of explicitly tracked objects. Therefore, offline 3D reconstruction or manual modeling can deliver largely error-free scene geometry, greatly simplifying photometric registration and radiance transfer computation. Arbitrary online geometric changes, such as a user's moving hand in the scene, are not supported.

Concerning photometric registration, one simplification is to assume that real world lighting is static, so photometric registration can performed in a pre-process [9, 18, 22]. Online illumination changes, such as a user casting a shadow on the scene, are not supported. Alternatively, online photometric registration can use an invasive light probe, such as an omnidirectional camera or reflective ball [14, 13, 22], placed in the scene, to replace computing photometric registration by direct measurement. Finally, global illumination effects can be restricted to virtual light sources [15], avoiding photometric registration altogether.

Previous work [7] aims at delivering photorealistic AR without restricting either scene geometry or scene lighting to be static or otherwise constrained. In particular, this work introduces a novel approach for *probeless* photometric registration by relying only on input from a common RGBD camera to compute both dynamic geometry [21] and the environment light considering occlusions. However, the need for simultaneous reconstruction, photometric registration and radiance transfer computation is not easily accommodated in real time. In particular, robust RT computation based on dense raytracing is versatile, but costly, and thus requires optimization [12].

The natural direction for optimizing RT computation is exploiting spatial and temporal coherence [25]. After all, neither geometry nor lighting are changing arbitrarily from moment to moment

*lgruber, *langlotz, ◇schmalstieg@icg.tugraz.at
†psen@ece.ucsb.edu, ‡holl@cs.ucsb.edu

in real world situations. However, probeless environments such as ours are too brittle for sparse sampling methods such as virtual point lights [14].

The topic of this paper is a novel RT approach for combined light estimation and rendering, which improves previous work by using adaptive sampling for exploiting spatial coherence and RT caching for exploiting temporal coherence. In particular this proposed approach combines subsampling in image space and in visibility space with robust caching method of visibility information. We evaluate our approach concerning performance, rendering quality and light estimation quality of shadows between real and virtual.

2 Related work

Most approaches for determining environment light in AR use artificial light probes such as a specularly [3] or diffusely [1] reflecting ball, a camera with fish-eye lens [14, 6, 13] or a rotating camera [5]. However, inserting light probes is not always feasible, and probless approaches are clearly preferred. For example, Madsen et al. [16] use an analytic sun/skylight model, but are limited to outdoor scenes. Jachnik et al. [9] reconstruct a surface light field for a simple planar scene and factorize the light field into a diffuse and specular component. The latter is used to estimate the environment light. However, their approach cannot deal with illumination changes in real time. The work of Yao et al. [27] recovers the environment light from a depth map but does not focus on any further rendering methods. The work discussed in this paper can be classified as probeless AR for indoor scenarios.

A large body of work has been published about extending basic local lighting for AR, for example, by supporting discrete directional light sources and shadows from real onto virtual objects and vice versa. These approaches are also sometimes called common illumination [10, 3] to distinguish them from true global illumination approaches. Examples for the latter are methods based on virtual point lights, as presented by Knecht et al. [14] and Lensing et al. [15]. The work by Lensing et al. is noteworthy, because it uses an RGBD cameras to compute real-time global illumination for changing scenes, as we do. However, they do not obtain any environment light. Kan et al. [13] increase realism by computing arbitrary specular and refractive effects with real-time raytracing. However, fully recursive raytracing is generally computationally expensive and was only shown for small scenes. The work by Nowrouzezahrai et al. [22] is similar to our radiance transfer approach based on spherical harmonics (SH), but they use a light probe and do not deal with fully dynamic scene geometry.

Ray-based sampling is expensive, in particular if multiple visibility tests and extra light bounces are desired per sample point. Therefore, accelerating ray-based sampling has been a frequent topic in computer graphics research. The key observation here is that rays are independent, and we can select the ones that contribute most. Because simple regular subsampling suffers from aliasing artifacts [19], anti-aliasing techniques such as stochastic sampling [4] and progressive refinement [23] were proposed, relying on spatial coherence to select samples. Alternatively, temporal coherence can be exploited by caching rendering results from frame to frame in an animated sequence [25]. One key idea we adopt here is that sampling can be guided by auxiliary geometry buffers if the rendering system can provide them.

3 Efficient and robust radiance transfer

3.1 Radiance transfer setup

We begin our discussion with an overview of the RT computation used by Gruber et al. [7], which is the object of the optimizations described in this paper. RT is based on the common reflection equation [11] and describes how light interacts with the environment. Our RT supports Lambertian reflection and shadows by testing the visibility for each surface point x. Since the *world space*

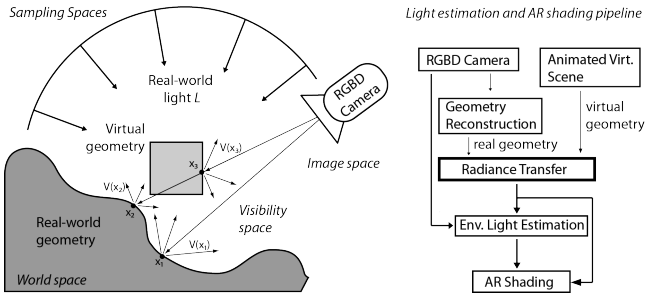

Figure 2: Overview of the different sampling spaces and radiance transfer setup. It shows how the rays are cast from the camera image space into the world space, where they hit a surface point x. From this point, visibility rays V are sent out to test for occlusions. The right figure shows the entire processing pipeline.

is given as a voxel model created by KinectFusion [21], the actual surface is represented implicitly as a truncated signed distance function (TSDF). We evaluate the TSDF by casting rays from *image space* into world space. For each surface point, we cast secondary visibility rays V into a spherical *visibility space* to test for against occlusions (cf. Figure 2). This results in a per pixel radiance transfer signal, which is compressed using spherical harmonics (SH). SH reduces the complexity of RT for the light estimation and employed for the final differential rendering [2]. Note that differential rendering demands computing two solutions per ray, as indicated by points x_2 and x_3. In the following we discuss our approach, which combines operation in world space, screen space and visibility space.

3.2 Joint image and visibility space subsampling

Since dense RT sampling is not computationally feasible, we resort to adaptive subsampling. We adopt different sampling regimes for image space and visibility space. Since we need to compute the visibility signal V only for currently visible surface points, we can express it as a 4D function $V(r,s,\phi,\theta)$, where a surface point x is projected onto a pixel (r,s) in image space, and (ϕ,θ) encode the direction of the visibility ray in polar coordinates. The joint image and visibility space subsampling can be divided into the following four major steps (see Figure 2).

Step 1 - Adaptive image space subsampling: Regular subsampling (every n^{th} pixel) of the scene geometry is required by the light estimation but introduces aliasing to visible virtual content. To improve rendering quality we add dense RT sampling to areas where interesting light interactions between virtual and real objects are expected. For virtual content, these are non-occluded pixels close to a depth or normal discontinuity, where we expect illumination gradients to occur. We determine these pixels by computing edges in the depth and normal buffer. For real content, pixels affected by virtual objects in differential rendering must be densely sampled. We determine these pixels by marking surface points hit by visibility rays starting at virtual objects.

Step 2 - Interleaved visibility subsampling: We exploit the limited rate of change in environment lighting by interleaved subsampling in visibility space. Since the visibility space is established relative to world space, and visibility is stored in a bandwidth-limited SH representation, the result is not as susceptible to error in tracking and noise in surface reconstruction, as would be the case with the more common spatially interleaved sampling. For every sample point, we evaluate only a subset of all ray directions per frame. More precisely, for every chosen pixel (r,s) and every k^{th} frame, we compute a different subset of N/k rays out of N possible ray directions. By interleaving the ray directions, each subset

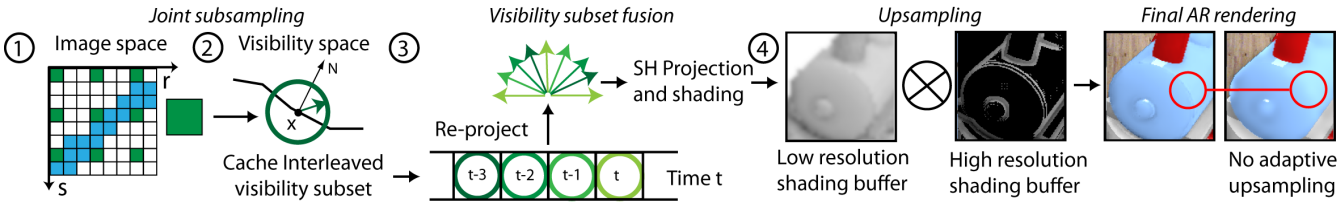

Figure 3: Starting from the left, we see the adaptive sampling pattern, consisting of regular subsampled (green) pixels and densely sampled pixels (blue). For all green pixels we compute an interleaved subset of visibility rays over time. The subsets are stored in a 2D cache and fused by re-projection for every frame. The complete RT signal is projected into SH for light estimation and shading. Due to regular subsampling, the shading map has a lower resolution than the final AR rendering. Therefore, we apply upsampling for the visible augmentations to increase the rendering quality. The visual improvement (highlighted by the red circles) can be seen in the right most part of the Figure.

evenly samples the entire visibility space. The visibility testing results of the current subset are then cached in an image space aligned 2D cache. After k frames, the previous subset of N/k rays is replaced by new results to refresh the cache. Overall, this results in k caches holding the complete visibility space. The next step explains how we recover the entire visibility space in every frame.

Step 3 - Reprojection and visibility fusion: Since the camera moves, the visibility caches are misaligned. To correctly access each of the k visibility caches for one current pixel, we reproject the associated 3D surface point into the camera coordinates associated with each 2D cache. Cache entry association after reprojection is based on a nearest neighbor lookup. The subsampling in image space used in step 1 leads to increased robustness of the reprojection, since we project from a higher resolution into a lower resolution 2D cache. As common with reprojection, the depth buffer from the previous frame is used to weed out disocclusions or dynamic scene changes. If a threshold depth difference for the sample point is exceeded, the cache entry for the sample point is discarded and rebuilt.

Step 4 - Upsampling: The final upsampling to full image resolution is performed in a deferred rendering step. Shading for areas with low resolution and high resolution RT sampling is computed separately and combined with the color buffer resulting from rendering unshaded virtual content over the video background.

3.3 Robust visibility state caching in world space

The world space is the natural space of the 3D reconstruction, which suggests to adopt it also for RT caching (per voxel). However, TSDF reconstruction relies on a probabilistic surface model, i. e., multiple consistent observations of a surface point are required to filter sensor noise, which makes it unreliable for caching RT directly. Since we want to handle dynamic scene changes, we cannot simply turn off TSDF updates.

A second characteristic of a TSDF volume compared to the polygonal models commonly used in real-time global illumination is that the spatial resolution is much lower than the useful 3D resolution at which the implicit surface can be sampled. This has the consequence that multiple pixels (r,s) in image space may be associated with a single TSDF voxel, especially if the camera is close to the surface. As a consequence, any information stored in world space per voxel is stable with respect to camera movement and tracking errors, but at the expense of low image space resolution. Again, storing RT per voxel in world space is not suitable, as it would suffer from the insufficient resolution.

We therefore built the RT cache in image space and combine it with a visibility state cache in world space. Visibility state is a flag that determines if a certain surface point is free from occlusion and therefore need not be considered for visibility testing.

Cache fill: The visibility state is evaluated statistically. If no more than a certain small fraction of the visibility rays is blocked, we assume that the surface point x is unoccluded and store this information. If we evaluate this surface point again in a subsequent frame, we compute only the Lambertian term without evaluating V. Since the maximum ray length considered for V is limited, this optimization is assumed to remain valid until the reconstruction changes in the neighborhood of the surface point. Only one Boolean is required to store the visibility state, so this cache has a negligible memory footprint.

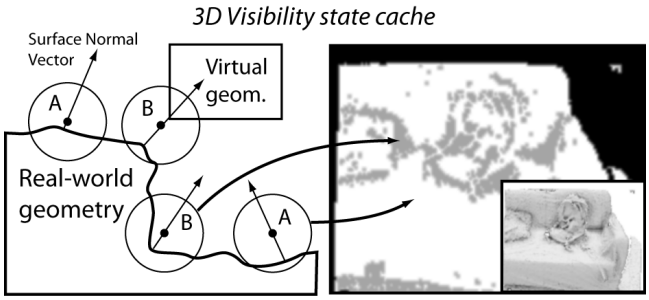

Figure 4: Left: Reconstructed surface area with noisy surface normals and virtual geometry. (B: occluded,A: non occluded). Right: Example of the visibility state cache (grey: occluded, white: non occluded). Right lower inset: shaded buffer of the real-world geometry.

Cache update: Updating the cache is more difficult, since it requires observing changes in 3D. This problem is attacked by approximating the 3D solution with multiple 2D projections. We achieve this by using one or more cache validation cameras (CVC), which obtain depth maps from raytracing into the TSDF. The depth map from a CVC is compared with a stored one to determine invalid cache entries, where the surface has changed. An invalid cache entry is simply removed to trigger resampling at the earliest convenience. CVC placement can be strategically chosen to cover all relevant changes in the volume. We usually place one CVC overhead for observing the whole volume at optimal resolution, and another one as a dynamic view-dependent camera following the user. Note that a dynamic camera must compute two depth maps (from the old and the new point of view) using a ping-pong buffering scheme.

4 Implementation Details

In the following we discuss implementation details. All time critical parts of the pipeline run on the GPU (NVIDIA GeForce 780).

Reconstruction of the scene geometry: For the purpose of scene reconstruction and camera tracking, we employ KinectFusion [20] implemented in CUDA. We use input from a RGBD cam-

era (Microsoft Kinect) at a resolution of 640x480, which is also the final image resolution. The dimensions of the working volume are 2x2x2 meters with a voxel resolution of 268^3.

Computation of the radiance transfer: For the visibility sampling, we used in total 81 ray directions and a maximum visibility ray length of 10 cm. We empirically determined a sample spacing in images space of 4 pixels and a temporally interleaved sampling interval of 4 frames.

Estimation of the environment light: Environment light is estimated in SH form using four bands (16 coefficients). The matrix solver combines CUDA with CPU computation, but could be ported to the GPU entirely. However, the performance impact is negligible.

Shading of the final image: The AR rendering pipeline is based on deferred shading implemented in OpenGL shaders.

5 Evaluation and Results

We evaluated performance, visual quality, and light estimation quality of our approach. The test data sets were motivated by typical requirements of AR applications such as home shopping (Figure 1, left).

We compare different parameterizations of our approach against a reference method with full sampling in image and visibility space ("RM") and regular subsampling in image space with a sample spacing of 4 ("REG4") and 8 ("REG8"). Joint subsampling methods have sample spacings in image space of 1 ("ART1"), 4 ("ART4") and 8 ("ART8"), while sample spacing was set to 4 in visibility space.

We use multiple real-world sequences recorded with the RGBD camera as test data sets for performance and rendering quality evaluation. The light evaluation requires some form of ground truth. Real-world lighting, such as obtained from a spherical mirror ball, would add additional errors. Therefore, we use synthetic light sources as ground truth. We substitute the input RGB image from the camera by renderings of the real-world geometry lit by the synthetic light sources. In total, we use five different sequences, each 300 frames long, containing dynamic camera movements and dynamically changing geometry. An overview of the scenes and an example for dynamically changing geometry is shown in Figure 9(a) and (c).

5.1 Performance

We evaluate performance by measuring the computation time of each single frame and report this in frames per second (FPS). The performance mainly depends on the radiance transfer computation, but other parts of the pipeline can have an effect, such as changing geometry. Figure 5 reports the FPS median over all test sequences. As expected, RM is the slowest method. ART1 gains performance speedup solely from interleaved sampling in visibility space. The values of the methods REG4 and REG8 demonstrate the upper bounds of performance gain through regular sampling in image space. The results of ART4 and ART8 are slightly faster than their regular counterparts REG4 and REG8, even though the ART methods compute adaptive upsampling to improve the visual quality. The impact of adaptive upsampling on performance can vary, since it is view dependent, leading to stronger variations in frame rate. However, ART4 shows a significant performance boost over RM.

5.2 Visual quality

Visual quality assessment is more complex than performance assessment and requires multiple metrics [26] for a complete analysis: Therefore, we compare the final rendered result from each method to RM using both absolute differences and HDR-VDP2 [17]. Absolute differences detect all deviations from the RM image, while

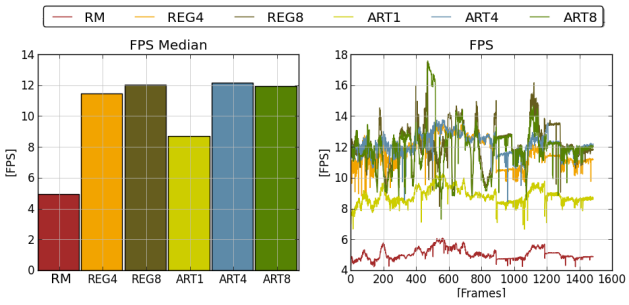

Figure 5: Median FPS (Left) and FPS (Right) over all frames. Timings in ms for RT and light estimation: RM 203.5, REG4 88.886, REG8 83.385, ART1 117.06, ART4 85.306, ART8 79.827

perceptually driven measurements such as HDR-VDP2 mainly rely on saliency detection, which focuses on contrast and brightness changes. We inspect the visual quality of the final AR image sequence frames along the following dimensions:

Image differences: As a first assessment, we investigate pixel-wise image differences compared to RM. We compute the median of the sum of absolute pixel differences (MSAD) over all frames (cf. Figure 7, Column 1). Column 2 in Figure 7 shows the normalized MSAD values as image quality (best quality is 100% corresponding to the RM output) vs. the median performance over all frames. These results correspond to the renderings (Figure 6, first row), where the ART methods subjectively show better results than the REG methods. Note that image differences were determined over the entire image, but only areas affected by virtual content or differential rendering can cause errors. The size of these areas varies and can be rather small, which makes the measurements very sensitive.

Image quality prediction: We compute the subjective mean opinion score Q_{mos} from HDR-VDP2, which predicts image quality as perceived by a human observer (cf. Figure 7, Column 3).

Visibility error probability: We also measured the maximum probability $P_{det} = max(P_{map})$ from HDR-VDP2 that a human observer would perceive any image difference between the reference image and the other methods (cf. Figure7, Column 4). In the second row of Figure 6, P_{map} is visualized as a heat map for one frame over all methods. In summary, the results imply that ART4 has comparable image qualities to the reference method.

5.3 Light estimation quality

Differences in the light estimation cause brightness and contrast differences in the final rendering. Therefore we examine the quality of the light estimation of each method relative to the reference method and propose two measurements. Note that we use synthetically generated light sources as ground truth. We shade the reconstructed geometry with this synthetic light and use the resulting renderings (Figure 9(a)) as simulated input for the photometric registration pipeline. The synthetic light source is created by directly projecting a low frequency function into SH coefficients. This low frequency function simulates an animated circular area light source, which falls off from center to border and has a distinguishable dominant light direction.

Environment map differences: As the environment map is a very common light source representation, we project the estimated light source from SH into a cubic environment map and evaluate the median of the pixel differences between the estimated and the reference environment light over all frames. (Figure 8).

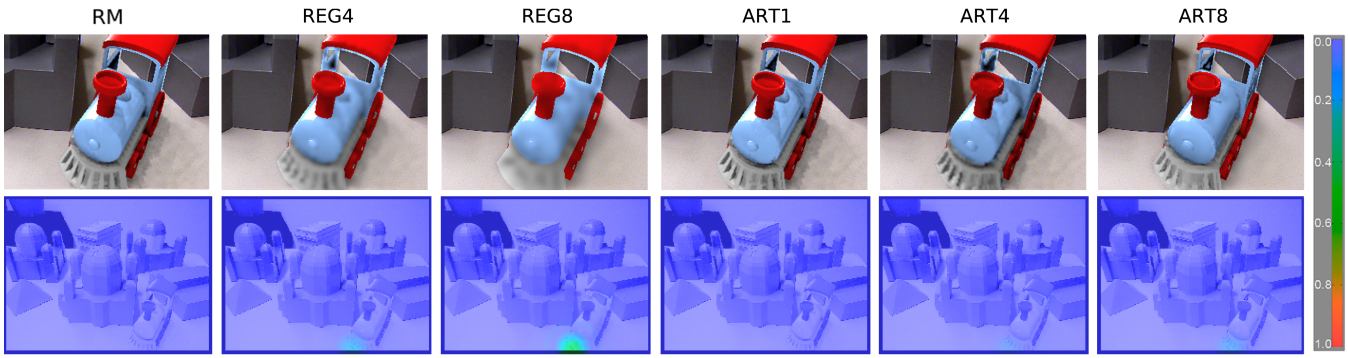

Figure 6: First row: close-up views of one AR test sequence. The methods solely sampling in image space (REG4 and REG8) loose details in the high frequency parts of the geometry due to naive upsampling. The joint subsampling methods incorporate adaptive upsampling, which preserves those critical areas. Second row: heat map visualization of the visibility error probability P_{det} (legend on the right of the Figure).

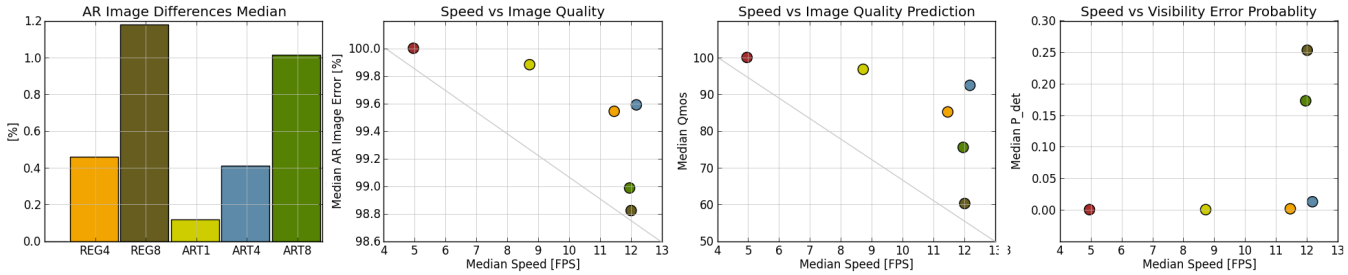

Figure 7: First and second column: median of absolute image differences (MAID) over all frames of the data set. The joint subsampling methods (ART1, ART4 and ART8) have smaller pixel differences to RM. In column 2, we express image difference as percentage of the image quality, where 100% is the best quality, for better comparison to the values in column 3 which show the results of the image quality prediction [17]. Note that the x-axis shows the median FPS. The best method has to appear in the upper right corner of the plot. Column 4: Speed vs. probability that an image error is visible to the user. The results indicate that ART4 has the best trade-off between speed and image quality.

Dominant light source: The dominant light direction can have a strong impact on local shading models and is hence a valuable measure. We estimate the dominant light source direction from the first three SH coefficients [22]. Then we compare the estimated dominant light source with the synthetic one and compute deviations as the dot product of the respective direction vectors (Figure 9(b)). The average and standard deviation of the dot product between the reference method and the estimated light over all frames are reported in Table 5.4 for two sequences. We observe a higher error for sequences with moving cameras. However, the absolute error of the reconstructed dominant light estimation is relatively small.

5.4 Rendering results

The employed radiance transfer setup for light estimation is particularly well suited for low-frequency lighting. However, many real world scenarios involve a mix of low-frequency and high-frequency light sources. By computing the dominant light source direction, we can add view dependent lighting such as Phong shading for virtual objects introducing glossy materials. Results with the mixture of diffuse SH based rendering and specularities are shown in Figure 1.

6 Conclusion and Future Work

In this work, we have presented an approach for efficient RT computation for real-time probeless light estimation and photorealistic rendering in AR. Our work combines suitable adaptive sampling and caching techniques to guarantee real-time performance, while supporting dynamic scenes and dynamic illumination. The results demonstrate that soft shadows and environment lighting effects are possible without restricting the environment or requiring unusual hardware. Our quantitative evaluation shows that the proposed approach significantly improves performance without degradation in image quality compared to previous work.

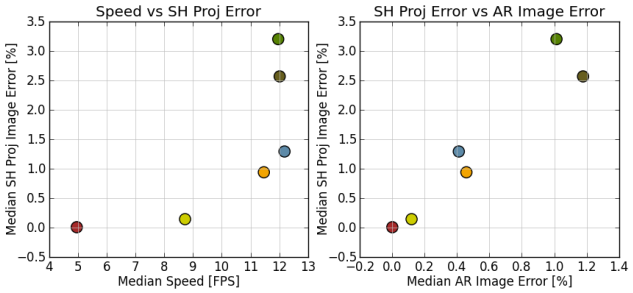

Figure 8: Left: the MSAD of the environment light maps. ART4 and ART8 score slightly worse then REG4 and REG8. Right: environment light maps MSAD vs. the AR rendering MSAD. A correlation between image quality and light estimation quality can be observed.

Acknowledgements

This work has been funded by the Christian Doppler Laboratory for Handheld Augmented Reality.

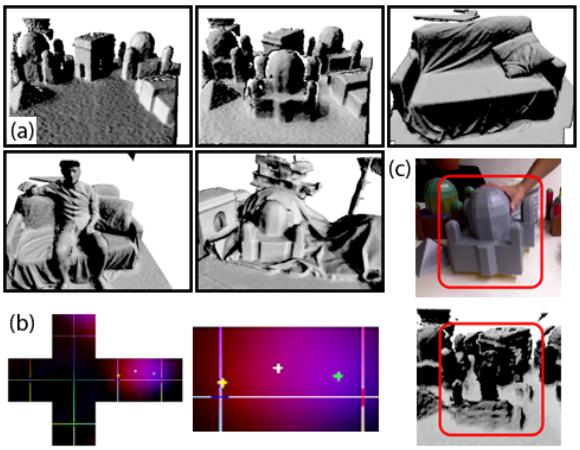

Figure 9: (a) Data set frames lit by synthetic light source. (b) Environment light map: estimated (red), synthetic (blue) and difference (violet). Dominant light directions: green (Synth. light ground truth), white (estimate) and current camera view direction (yellow). (c) Test scene with dynamically inserted objects (red rectangle). Top: RGB input video. Bottom: reconstructed volume. The reconstruction lags behind by a couple of frames. This naturally results in a misalignment of the radiance transfer and the reflection observed from the camera image.

	RM	REG4	REG8	ART1	ART4	ART8
Avg	0.9488	0.9399	0.9393	0.9574	0.9475	0.9521
Std	0.0348	0.0446	0.0449	0.0320	0.0412	0.0376
Avg	0.9616	0.9606	0.9626	0.9621	0.9559	0.9624
Std	0.0196	0.0261	0.0257	0.0186	0.0302	0.0256

Table 1: Average and standard deviation of the dot product between the estimated and the synthetic dominant light source directions. First row: static camera path. Second row: dynamically changing camera path.

REFERENCES

[1] M. Aittala. Inverse lighting and photorealistic rendering for augmented reality. *The Visual Computer*, 26(6-8):669–678, Apr. 2010.

[2] P. Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 189–198. ACM, 1998. ACM ID: 280864.

[3] P. Debevec, A. Wenger, C. Tchou, A. Gardner, J. Waese, and T. Hawkins. A lighting reproduction approach to Live-Action compositing. Technical report, July 2002.

[4] M. A. Z. Dipp and E. H. Wold. Antialiasing through stochastic sampling. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '85, page 6978, New York, NY, USA, 1985. ACM.

[5] S. DiVerdi, J. Wither, and T. Höllerer. Envisor: Online Environment Map Construction for Mixed Reality. In *Proceedings of the IEEE Virtual Reality Conference 2008*, pages 19–26. IEEE, 2008.

[6] T. Grosch, S. Mueller, and W. Kresse. Goniometric light reconstruction for augmented reality image synthesis. In *Proc. GI Jahrestagung*, Frankfurt, Germany, 2003.

[7] L. Gruber, T. Richter-Trummer, and D. Schmalstieg. Real-time photometric registration from arbitrary geometry. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 119 –128, Nov. 2012.

[8] R. Herzog, E. Eisemann, K. Myszkowski, and H.-P. Seidel. Spatio-temporal upsampling on the gpu. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D) 2010*. ACM, 2010.

[9] J. Jachnik, R. A. Newcombe, and A. J. Davison. Real-time surface light-field capture for augmentation of planar specular surfaces. In *ISMAR*, pages 91–97, 2012.

[10] K. Jacobs and C. Loscos. Classification of illumination methods for mixed reality. *Computer Graphics Forum*, 25(1):29–51, Mar. 2006.

[11] J. T. Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '86, page 143150, New York, NY, USA, 1986. ACM.

[12] P. Kan and H. Kaufmann. Differential irradiance caching for fast high-quality light transport between virtual and real worlds. In *Proceedings IEEE ISMAR 2013*.

[13] P. Kan and H. Kaufmann. High-quality reflections, refractions, and caustics in augmented reality and their contribution to visual coherence. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 99–108, Nov.

[14] M. Knecht, C. Traxler, O. Mattausch, W. Purgathofer, and M. Wimmer. Differential instant radiosity for mixed reality. In *2010 9th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 99–107. IEEE, Oct. 2010.

[15] P. Lensing and W. Broll. Instant indirect illumination for dynamic mixed reality scenes. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 109–118, Nov.

[16] C. Madsen and B. Lal. *Probeless Illumination Estimation for Outdoor Augmented Reality*. INTECH, 2010.

[17] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich. HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Trans. Graph.*, 30(4):40:140:14, July 2011.

[18] M. Meilland, C. Barat, and A. Comport. 3d high dynamic range dense visual slam and its application to real-time object re-lighting. In *Proceedings IEEE ISMAR 2013*.

[19] D. P. Mitchell. Generating antialiased images at low sampling densities. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, page 6572, New York, NY, USA, 1987. ACM.

[20] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1498–1505. IEEE, June 2010.

[21] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: real-time dense surface mapping and tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, page 127136, Washington, DC, USA, 2011. IEEE Computer Society.

[22] D. Nowrouzezahrai, S. Geiger, K. Mitchell, R. Sumner, W. Jarosz, and M. Gross. Light factorization for mixed-frequency shadows in augmented reality. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, page 173179, Washington, DC, USA, 2011. IEEE Computer Society.

[23] J. Painter and K. Sloan. Antialiased ray tracing by adaptive progressive refinement. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '89, page 281288, New York, NY, USA, 1989. ACM.

[24] D. Scherzer, L. Yang, O. Mattausch, D. Nehab, P. V. Sander, M. Wimmer, and E. Eisemann. A survey on temporal coherence methods in real-time rendering. In *In State of the Art Reports Eurographics*, 2011.

[25] D. Scherzer, L. Yang, O. Mattausch, D. Nehab, P. V. Sander, M. Wimmer, and E. Eisemann. Temporal coherence methods in real-time rendering. *Computer Graphics Forum*, 31(8):2378–2408, 2012.

[26] M. Čadík, R. Herzog, R. Mantiuk, K. Myszkowski, and H.-P. Seidel. New measurements reveal weaknesses of image quality metrics in evaluating graphics artifacts. *ACM Trans. Graph.*, 31(6):147:1–147:10, Nov. 2012.

[27] Y. Yao, H. Kawamura, and A. Kojima. Shading derivation from an unspecified object for augmented reality. In *2012 21st International Conference on Pattern Recognition (ICPR)*, pages 57–60, 2012.