Quality Scalability of Soft Shadow Mapping

Michael Schwarz Marc Stamminger

University of Erlangen-Nuremberg

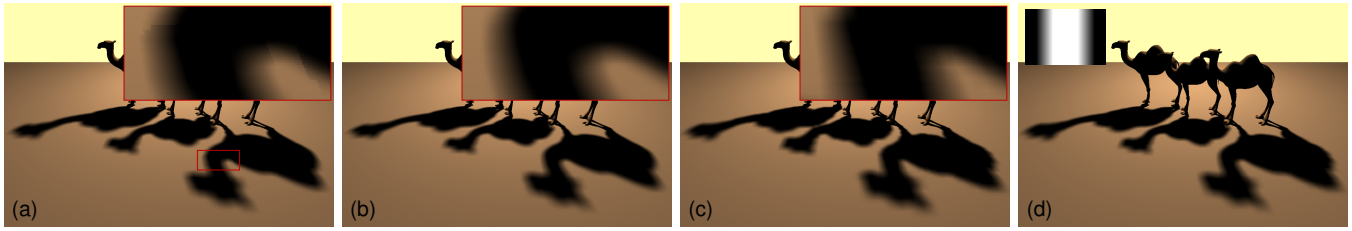

Figure 1: (a) Soft shadow mapping (micro-occluder budget $b_{\text{mo}} = 12$) without blending can lead to transition artifacts when varying shadow map levels need to be used across pixels. (b–d) Our scheme enables not only smooth transitions for screen-uniform budgets (b: $b_{\text{mo}} = 12$, c: $b_{\text{mo}} = 5$) but also readily supports local budget variations (d: b_{mo} decreases from 12 in the center to 5 at the left and right borders as shown in the inset).

Abstract

Recently, several soft shadow mapping algorithms have been introduced which extract micro-occluders from a shadow map and backproject them on the light source to approximately determine light visibility. To maintain real-time frame rates, these algorithms often have to resort to coarser levels of a multi-resolution shadow map representation which can lead to visible quality degradations. In particular, discontinuity artifacts can appear when having to use different shadow map levels across pixels.

In this paper, we discuss several aspects of soft shadow quality. First, we motivate and propose a scheme that allows for varying soft shadow quality in screen-space in a visually smooth way and also for its adaptation based on local features like assigned importance. Second, we suggest a generalization of micropatches which yields a better occluder geometry approximation at coarser shadow map levels, thus helping to reduce occluder overestimation. Third, we introduce a new hybrid acceleration structure for pruning the search space of potential micro-occluders that enables employing finer shadow map levels and hence increasing quality. Finally, we address multisampled rendering and suggest a simple scheme for interpolating light visibility that only adds a negligible cost compared to single-sample rendering.

Keywords: soft shadows, level of quality, soft shadow mapping

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1 Introduction

Soft shadows are an important global illumination effect and have been the focus of many algorithms. The underlying problem of determining the visibility of an extended light source from each point in a scene constitutes a challenging and computationally demanding task. To obtain real-time performance, current algorithms have to resort to impose constraints on the scene that enable simplifications and precomputations, or to introduce approximations.

In the latter case, the quality of the rendered soft shadows is affected by the kind and degree of the approximation, with its associated cost influencing frame time. Often, the necessity or desire arises to vary soft shadow quality locally or temporally, for instance to adapt the frame rate. To avoid artifacts, resulting quality transitions should be smooth.

Apart from increasing the approximation budget, soft shadow quality can also be improved by choosing the approximations to be as faithful to the original as possible. It is further important to avoid useless computations and spent the available time only on those parts which actually affect the result, since this allows to take the finest-degree approximation possible for a given time budget and thus to get the best quality. For instance, it is reasonable to only consider those occluders which actually influence the light visibility of a shadow receiving point. Furthermore, with multisampled rendering becoming ubiquitous, the challenge arises to provide soft shadows in comparable quality to single-sampled settings without significantly increasing frame time.

In this paper, we address all four of the aforementioned aspects influencing soft shadow quality in the context of soft shadow mapping algorithms. In particular, after providing a detailed discussion of general possibilities to smoothly vary soft shadow quality in screen-space, we introduce a practical scheme to achieve such smooth changes which allows locally adapting the quality according to visual importance or high texture masking (Section 4). We further present a new micro-primitive which provides a better occluder approximation at coarser scales than previous approaches (Section 5). To better concentrate the computational resources on the relevant occluders, we suggest a new hybrid acceleration structure for search area pruning which features a low overhead (Section 6). Finally, we tackle the important multisample issue and introduce a simple interpolation scheme which only incurs a minor additional cost compared to single-sample rendering while retaining soft shadow quality (Section 7).

2 Related work

As a comprehensive treatment of soft shadow algorithms is beyond the scope of this paper, we restrict ourselves to review a small subset of the newer ones and refer to Hasenfratz et al. [13] for a survey of older approaches.

One class of algorithms operates in object space, with soft shadow volumes [2] being the major representative. Recently, Eisemann and Décoret [8] introduced a bit-field-based visibility algorithm that loops over all blocker triangles and applied it to soft shadows. While accurate, only planar and bumpy shadow receivers without self-shadowing are supported.

The majority of real-time soft shadow algorithms, however, are image-based. They hence not only scale better with scene complexity but also support non-triangular geometry created in the fragment stage via fragment kills or z-value modifications. Like most of these algorithms, Arvo et al. [1] start with a shadow map. They utilize it to derive umbra regions to which they then successively add outer penumbrae with a modified flood-fill algorithm. In contrast, Fernando [9] samples the shadow map to search for blockers and derive an average blocker depth which is then used to compute a kernel size for percentage-closer filtering [19].

Lately, soft shadow mapping algorithms [3–5, 11, 12, 21] which employ a shadow map for constructing occluder approximations received much attention. Since various aspects of their soft shadow quality, like its smooth spatial variation, are the main focus of this paper, they are discussed in more detail in the next section.

Finally, some algorithms adopt a hybrid approach and render auxiliary geometry attached to extracted silhouettes, enriching the data of a standard shadow map [7].

3 Soft Shadow Mapping Basics

Soft shadow mapping (SSM) is a rather fuzzy term which lately has mainly be used for the class of algorithms which take a shadow map of the scene, extract an approximation of the occluder geometry from the shadow map, and for each pixel determine light visibility by projecting the occluder approximation onto the light source, aggregating occluded light areas [3–5, 11, 12, 21]. In the following, we only consider receiver-driven SSM algorithms which loop over all (relevant) occluder approximations in the fragment shader.

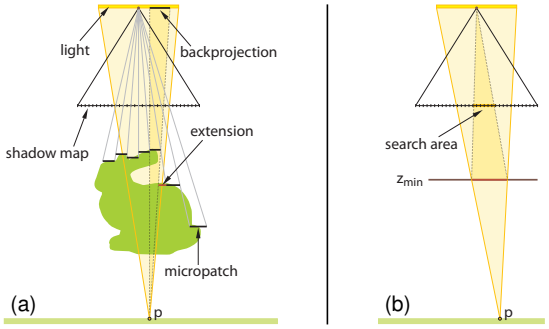

Figure 2: (a) Basic overview of soft shadow mapping. (b) Setup for search area refinement.

A simple occluder approximation is given by the shadow map texels unprojected into world space (cf. Fig. 2 a). Each resulting *micropatch* can then easily be depth-tested against a receiving point $\mathbf{p}$ and backprojected onto the light plane to derive the light area occluded by it. Depending on $\mathbf{p}$, gaps in the occluder reconstruction may emerge which lead to visible light leaks. This can be alleviated by extending a micropatch to its left and bottom neighbors in texture space [11]. Another kind of micro-occluder, the *micro-quad* [21], is obtained by unprojecting the shadow map texel centers and taking them as vertices of a quad mesh. Finally, it has been suggested to detect *occluder contours* from micropatches and perform radial area integration [12].

Usually, the light visibility is determined by simply accumulating the areas occluded by the single micro-occluders. Since backprojections may overlap, over-occlusion artifacts can appear. This can be avoided by tracking the visibility of light sample points with an occlusion bitmask [21].

For performance reasons, only relevant micro-occluders should be processed. A first estimate of the corresponding shadow map *search area* is given by intersecting the near plane with the point–light pyramid. If the depth range $[z_{\min}, z_{\max}]$ of the samples within the search area is known, an iterative refinement of the search area is possible by intersecting the plane $z = z_{\min}$ with the pyramid and projecting the result onto the near plane (see Fig. 2 b). Note that knowing the search area's depth range allows to identify pixels in umbra ($p_z > z_{\max}$) and completely lit ($p_z \leq z_{\min}$) regions where no micro-occluders need to be processed at all.

To determine the depth range of a shadow map area, an acceleration structure can be used. Both a *hierarchical shadow map* (HSM) [11], which is basically a min/max pyramid of the shadow map, and a *multi-scale shadow map* (MSSM) [21], which for each shadow map texel stores the depth ranges of all power-of-two-sized neighborhoods, have been proposed. The resulting multi-resolution representations of the shadow map also allow imposing an upper bound $b_{\text{mo}} \times b_{\text{mo}}$ on the number of micro-occluders considered during visibility determination. To meet this budget, micro-occluders may not be constructed from the original shadow map (level 0) but from coarser shadow map levels $i > 0$.

4 Smooth Variation of Soft Shadow Quality

Since soft shadows are expensive to compute, in practical applications, situations can arise where a lower soft shadow quality is accepted in some regions of the scene than in others, for instance to attain real-time frame rates. However, to avoid visual artifacts, the quality should smoothly be varied spatially in such settings.

In case of SSM, one way to adapt quality is by choosing the shadow map level used for micro-occluder construction. This quality variation is already in wide use to satisfy a user-specified upper bound $b_{\text{mo}} \times b_{\text{mo}}$ on the number of considered micro-occluders which is necessary to ensure real-time frame rates. Since the used shadow map level can vary across the pixels in a soft shadow region, transition artifacts may appear (cf. Fig. 1 a). Hence note that the practical employment of current SSM algorithms also mandates the ability to smoothly vary the quality because manually choosing an appropriate micro-occluder budget that renders transitions unnoticeable cannot really be considered a viable alternative.

Before presenting our approach in Subsection 4.2, we first take a more general look at levels of quality for soft shadows to explore the solution space and provide motivation for the choices finally made.

4.1 Level of quality for soft shadows

To meet a given frame time, a common approach taken for geometry is to resort to a coarser level of detail (LOD) [17], trading visual quality for speed. Most flexibility and adaptability is offered by view-dependent LOD schemes [14, 16] where the geometric detail can locally be changed, allowing for finer detail at silhouettes without having to apply the corresponding LOD to the whole model. The transition between two LODs is usually smoothed either by some form of alpha-blending or via geomorphing—at least when the difference would be visible otherwise.

Motivated by these techniques for geometry, in the following we discuss possible approaches to adapt the level of quality (LOQ) when rendering soft shadows. An important difference to geometric LOD is that soft shadow LOQ cannot be dealt with on a per-object basis but is defined in image space. Moreover, we cannot leverage offline pre-processing steps; at least not if we don't want to impose any restrictions on the occluders or to require a strict shadow caster/shadow receiver partition of the set of objects.

Multiple algorithms of different quality One option to tackle soft shadow LOQ is switching the soft shadow algorithm, i.e. each LOQ is defined by a distinct algorithm. Unfortunately such a scheme would only allow for discrete LOQ since a smooth transition between the results of different soft shadow algorithms is usually not feasible because of incompatible assumptions, simplifications and approximations. For instance, faking soft shadows by just smoothing depth comparison results with percentage-

closer filtering (PCF) [19] might seem a good candidate for a lower LOQ. Assuming SSM constitutes the next higher LOQ, it becomes important for the PCF approach to spatially vary the filter kernel size to account for non-uniform penumbra widths. While filtering with screen-space-local kernel sizes is possible at high frame rates [15], reliably determining the required kernel size is still an open research problem. A simple heuristic [9] exists, however it requires many shadow map samples in an occluder search step and is far from robust. But even ignoring these issues, a smooth transition to SSM wouldn't be possible in general because the PCF approach only takes into account the result of depth comparisons but not whether the used samples actually occlude the light or get projected next to it. Also the implicit assumption that the fraction of the light source that is occluded by a sample is given by its PCF filter weight is incompatible with micro-occluder approximations. Because of these and many further problems, we will only consider LOQ schemes using a single algorithm in the following.

Geometric occluder LOD When operating directly on the occluder geometry, another option to provide soft shadow LOQ is to apply a geometric LOD scheme to the occluders. Ultimately, for each pixel in a soft shadow region a separate view-dependent LOD of the occluder geometry that is consistent across neighboring pixels is desired. Since providing this pixel-wise LOD would incur a tremendous cost, in practice simpler schemes have to be adopted. Unfortunately, this is only reasonable if objects are classified as either shadow casters or shadow receivers since otherwise problems exist with self-shadowing and when a caster and a receiver are in contact.

Sparse visibility sampling Because light visibility often varies rather smoothly in penumbra regions, image-based soft shadow algorithms offer the possibility to adapt LOQ by performing the visibility evaluation not per pixel but according to a sparser sampling with a subsequent interpolation step. However, it is unclear whether smoothly varying the relative sparseness parameter (sparseness relative to (spatially varying) sample density required for a certain quality) suffices to allow for smooth LOQ transition or whether blending between the results obtained for two different sampling sparsenesses is required. Guennebaud et al. [12] successfully implemented a sparse sampling scheme for their SSM algorithm, enabling high speed-ups. But since they are using a fixed relative sparseness parameter value (called $s_{\min}$) throughout the screen, the LOQ can only be influenced globally. We also note that with increasing sparseness objectionable patterns can appear with their approach. These can be expected to become particularly noticeable in animated scenes because the underlying sparse sampling pattern is fixed in screen-space.

Intrinsic algorithm parameters Another option for realizing soft shadow LOQ is to vary an intrinsic parameter of an algorithm that affects quality. In case of SSM, varying the bit field size and adapting the complexity of the bitmask's sampling pattern (completely regular, regularly jittered, or random) are possible choices when using occlusion bitmasks. Again, smooth LOQ transitions would require alpha blending. Employing varying numbers of depth maps or switching between area accumulation and occlusion bitmasks also constitute quality alteration parameters which however pose problems similar to those in multi-algorithm LOQ schemes concerning blending.

The most promising parameter is the shadow map level and hence the virtual shadow map resolution used, which is somewhat related to the geometric LOD of the occluders. Ideally, we want a hierarchical occluder representation derived from the shadow map which allows for smoothly varying soft shadow LOQ by blending between two hierarchy levels and computing light visibility for the resulting intermediate occluder representation instead of having to blend the visibility results for the two discrete hierarchy levels.

The microquad interpretation might seem like a natural candidate because it can be considered a geometry image [10] of the occluder/receiver surface as seen from the light that trivially supports geomorphing. However, it turns out that deriving coarser level vertices cannot be done by mere subsampling but must perform some kind of minimum aggregation of the light depth values. The major obstacle which prevents microquads (and supposedly any occluder representation) from achieving the desired ideal is the binary decision of whether they get backprojected or not. A microquad is only considered if all four vertices are closer to the light source than the shadow receiving point **p**. Therefore, while geomorphing between two levels causes a smooth transition of the involved microquads, once a vertex crosses the depth level of **p**, the corresponding microquad abruptly becomes included or excluded, respectively, from backprojection, i.e. the LOQ is not transitioning smoothly. Consequently, adapting the shadow map level requires alpha blending for smooth LOQ variations.

4.2 Practical soft shadow mapping quality variation

Summarizing the observations from the previous subsection, as unsatisfying as it may be, alpha blending appears to be unavoidable when smooth shadow quality variations are required. And as mentioned earlier, constraints on the frame rate do mandate using various shadow map levels across the screen in current SSM algorithms. As a consequence, we concentrate on adapting the soft shadow quality by varying the used shadow map level and on attaining smooth spatial transitions via alpha blending.

The main challenge is to derive a fractional shadow map level ℓ that varies continuously across a soft shadow region, with the fractional part driving the alpha blending. In a previous approach, Guennebaud et al. [12] suggest deriving ℓ from a continuously varying estimate of the closest occluder's depth obtained by linearly filtering the HSM. They hence implicitly assume that neighboring fragments sample the same HSM level when determining the depth range during search area refinement. In general, however, this assumption is not true and fails in particular if a more accurate acceleration structure like the MSSM is used. As a consequence, discontinuities concerning ℓ are introduced which can manifest themselves as visible artifacts despite alpha-blending.

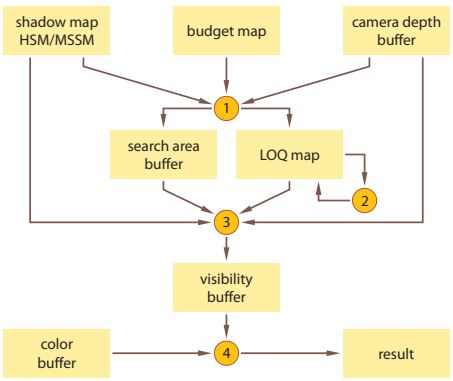

Figure 3: Overview of our approach for smooth soft shadow quality variation. Taking the local budget into account, we first determine the search area and the (fractional) shadow map level ℓ (1). We then smooth the LOQ map (2) and determine visibility, performing alpha-blending if necessary (3). Finally, the visibility value is applied to the color buffer (4).

Our approach to determine ℓ is summarized in Fig. 3 and tackles not only these shortcomings but also allows for adapting the shadow quality according to screen-space-local features like high texture masking or varying importance as assigned by the user or derived

by perceptually motivated algorithms. To this end, a budget map is provided as input which stores for each pixel (x,y) the maximum number $b_{\mathrm{mo}}(x,y)$ of micro-occluders considered along each light axis. By spatially varying the budget $b_{\mathrm{mo}}(x,y)$, soft shadow quality can be locally increased or decreased.

In a first step, we determine for each pixel by means of an acceleration structure like the HSM whether it is in umbra or completely lit and if not, the relevant search area for micro-occluders. Taking the pixel's budget $b_{\mathrm{mo}}(x,y)$ into account, we then derive an appropriate fractional shadow map level ℓ and store it in a texture referred to as LOQ map. Since the determined ℓ values don't necessarily vary smoothly across the screen, in a next step we apply a smoothing filter that respects geometric discontinuities. For performance reasons, we resort to a variant of separable bilateral filtering [18] where the smoothing kernel stops at depth discontinuities and at the interface to regions where no backprojection is required.

Subsequently, using the previously determined search area and the filtered ℓ value from the LOQ map, we perform the actual backprojection to determine light visibility. Depending on the blend weight $\alpha(\ell)$, the visibility may actually be determined twice—once for each of the closest integer ℓ values—and than linearly interpolated according to $\alpha(\ell)$. Finally, the resulting visibility value is applied to the color buffer.

Discussion Since alpha-blending is expensive as micro-occluders are not only backprojected for level $\lceil\ell\rceil$, satisfying the budget b_{mo}, but also for level $\lfloor\ell\rfloor$, violating the budget, it is reasonable to keep the transition region small. For instance, blending within a region of $\Delta\ell = 0.1$ has been suggested for practical use [12]. However, our experience shows that while this might be acceptable in static images, the resulting thinner blending regions often become visible in animated settings. Furthermore, severe artifacts can show up in regions where transitions between multiple levels occur within a very small neighborhood if the blending region is chosen too thin. Hence, for maximum robustness, we always perform alpha-blending. Consequently, a good classification of umbra and completely lit regions becomes even more important than before, i.e. using either the MSSM or the YSM presented in Section 6 is highly advisable.

Similar to aliasing in standard shadow mapping, depending on the scene configuration and the value of ℓ, the footprint of a single micro-occluder's influence region in screen-space may be large, resulting in jaggy-like shadow boundaries. Even worse, this large footprint can lead to swimming artifacts, that is, discontinuities in the temporal domain can appear once objects move relative to the light source and hence their rasterization in the shadow map changes. Apart from increasing the effective shadow map resolution, temporal smoothing could be applied to alleviate such artifacts, e.g. with a history buffer approach [20].

5 Better occluder approximation at coarser shadow map levels

In regions of larger ℓ values where coarser shadow map levels get used, overall soft shadow quality can be improved without decreasing ℓ by coming up with a better approximation of the occluding geometry at such coarse levels. Before presenting our new occluder approximation, we first briefly review existing ones.

5.1 Shadow-map-based occluder approximations

In the following we restrict ourselves to micropatches with extensions [11], microquads [21] and occluder contours [12]. First note that all of these approximations assume that occluder samples adjacent in either x or y direction belong to the same macro-occluder surface, i.e. no holes exist where light can pass through.

At the finest shadow map level $i = 0$, microquads probably provide the best fit to the occluder geometry, but tend to be slowest and suffer from underestimating the occluders. (Depending on the used algorithm, the resulting light occlusion may actually be overestimated if the microquads' projections on the light source overlap; but this is true irrespective of the chosen approximation.) In contrast, micropatches usually lead to occluder overestimation. Contours are in-between because they reduce overestimation and might even introduce underestimation. On the other hand, since contours are extracted in 2D instead of 3D space, occluders recorded in the shadow map may be missed.

The tendency to over- or underestimate occluders, respectively, is usually preserved at coarser shadow map levels, obtained by simple minimum aggregation. Fine structures can be missed by both microquads and occluder contours (if they are shrunk in coarser levels as suggested by Guennebaud et al. [12]) whereas micropatches implicitly enlarge them with each coarser level. We also observe that micropatches (having a uniform size in texture space) and microquad vertices (being uniformly spaced in texture space), respectively, often don't represent the occluder samples from the shadow map well at coarser levels. Since occluder contours are derived from micropatches, they also suffer from this problem.

5.2 Microrects as a generalization of micropatches

Central to better represent the occluders at coarser levels is the introduction of more flexibility in specifying a micro-occluder's extent. Therefore, we suggest generalizing micropatches by abandoning the limitation of uniform size in texture space. We pick micropatches as starting point because they are rather conservative concerning occlusion, not missing fine structures. Note also that occluder contours could be extracted from the resulting new micro-primitive, which we call *microrect*.

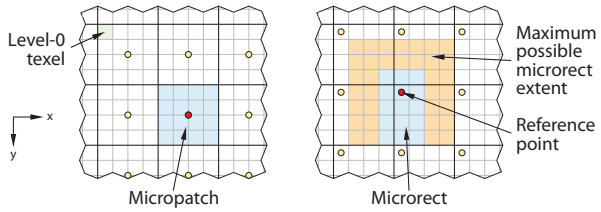

Figure 4: Exemplary level-2 micropatch (left) and microrect (right).

Each texel of shadow map level i completely defines a micropatch: it has a texture space reference point (the texel center), a fixed size corresponding to one level-i texel (or equivalently $2^i \times 2^i$ level-0 texels), and an associated light depth value. It represents the occluders captured by the related $2^i \times 2^i$ level-0 texels.

Microrects maintain this $1:1$ relationship between texels and micro-occluders. However, at coarser levels $i > 0$, their texture space extent is no longer restricted to a single level-i texel but can be any rectangular region of level-0 texels subject to the following two constraints: First, it must contain the microrect's reference point, which is obtained by subsampling the reference points from level $i-1$. Second, the extent may not contain the reference point of any other level-i microrect, thus imposing a maximum on it as illustrated in Fig. 4. These two requirements ensure that an acceleration structure like the MSSM can readily be used. On the other hand, since the microrects of levels $i > 0$ can have varying extents as well as overlap, both their associated depth values and their extents no longer can directly be obtained from the acceleration structure. We hence store them in two additional textures with full mip-map chains, i.e. the gained flexibility comes along with an increased memory footprint.

By construction, the extent and associated depth value have to be determined explicitly. To simplify the presentation, we first consider the 1D case. Since microrects at the finest level $i = 0$ are just micropatches, their extent and depth is trivially given by the corre-

sponding level-0 texel. For the remaining levels $i > 0$, the microrect extents and depth values are derived iteratively from the ones of the next finer levels $i-1$. Remember that the reference points of the level-i microrects are obtained by regularly subsampling the ones of the microrects from level $i-1$, as shown in Fig. 5. Consequently, we merge every second microrect (e.g. microrect B in Fig. 5) from level $i-1$ with one of its neighboring microrects (A and C) to derive the level-i microrects (M and N). We always pick that neighbor as merge partner which features the least depth value difference (here: C). The extent of the resulting level-i microrect (N) is given by the union of the extents of the involved finer-level microrects (B, C); its depth is derived by taking the minimum involved depth value.

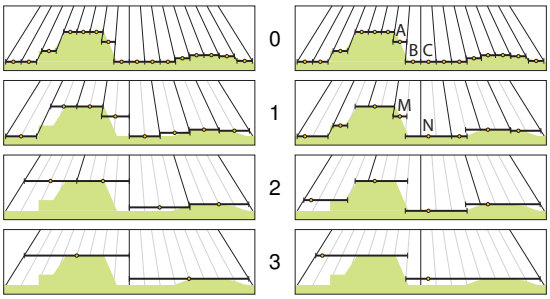

Figure 5: Micropatches (left) and microrects (right) at levels 0 to 3.

Unfortunately, the 2D case is much more involved since a microrect has neighbors in x, y, and diagonal x–y direction but a rectangular extent. Therefore, it is in general not possible to merge neighboring microrects such that the union of their extent is still rectangular. To avoid missing any occluder, we hence take the smallest rectangular region encompassing the merged extents as new extent. Note that this can lead to microrects with overlapping texture space extent. However, disallowing overlaps at all puts a heavy constraint on the microrects' ability to approximate the occluders because many potential microrect merges become prohibited. Consequently, we don't try to avoid overlaps but alleviate their influence on light occlusion by using occlusion bitmasks instead of area accumulation during light visibility determination. Note that this is often reasonable anyway when resorting to coarser levels for micro-occluder construction.

Construction To derive microrects in level $i+1$ from those in the next finer level i, we basically proceed like in the 1D case for determining the merge partners in x and y direction and treat the diagonal neighbors separately. As before, the depth associated with a microrect is obtained by minimum aggregation. More precisely, we suggest the following simple and fast greedy approach well-suited for data-parallel processing that keeps redundant comparisons to a minimum and proceeds in two passes. First, we consider merging the microrects retained after subsampling (e.g. microrect A in Fig. 6, which becomes M in level $i+1$) with their neighbors in positive x and y direction (B, D). The decision whether to merge is again based on the depth value differences between the merge candidate and its two involved neighbors (B–A, B–C; D–A, D–G). If the resulting extent completely contains the diagonal neighbor's one (E), we further merge with this microrect. In the second pass, we then merge with all those level-i neighbors in negative direction (E, F, H in case of P) which have not been aggregated yet in the first pass.

We note that a global optimization as well as not only taking samples from the next finer but from the finest level may yield better fits than our greedy approach, however real-time construction becomes rather difficult with more involved methods. Also note that a third pass which locally reduces overlaps is not an option since apart from the extent, the depth obtained via minimum aggregation would also have to be updated.

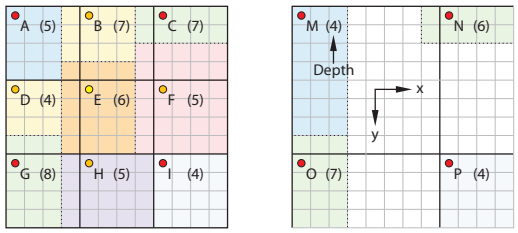

Figure 6: Setup for greedy microrect construction. Each uniformly colored rectangular region corresponds to a microrect, with the contained dot marking its reference point. Left: Input microrects at level $i = 2$. Right: Output microrects at level $i+1$ after the first pass.

6 Hybrid Acceleration Structure for Search Area Pruning

If an upper limit $b_{\mathrm{mo}} \times b_{\mathrm{mo}}$ on the number of micro-occluders considered during light visibility determination is imposed, which is usually the case, it is highly desirable to only pick micro-occluders that actually occlude some part of the light in order to get the best possible quality for the given budget. In practice, this translates to requiring the employed acceleration structure to yield a tight search area of potentially relevant micro-occluders. Moreover, for better overall performance, the acceleration structure should detect for as many pixels in umbra or completely lit regions as possible that they don't require micro-occluder backprojection at all.

While the MSSM [21] is an attractive candidate because it yields tight search areas and very good classification results of umbra and fully lit regions, both its construction and its memory footprint become too expensive for shadow map sizes greater than 1024^2 (cf. Table 1). On the other hand, because of its pyramidal nature, the HSM [11] entails significantly lower costs but also usually offers much more conservative results.

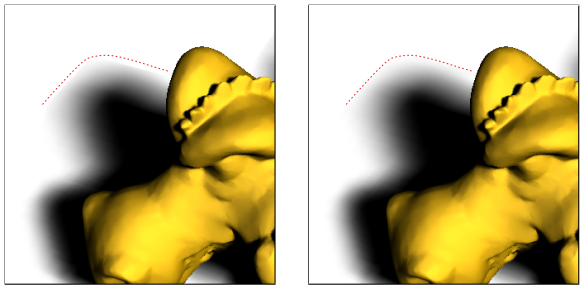

Figure 7: Using the MSSM may lead to "over-precision" artifacts at the interface to fully lit regions (left) which can be avoided by adapting the region queried during search area pruning appropriately (right). The red dashed line indicates the actual interface introduced by the coarser-level micro-occluders that get backprojected.

"Over-precision" artifacts A subtle aspect when using the MSSM that is not explicitly addressed in our previous paper [21] is that unless the area queried for its depth range is grown to account for the current budget, resulting classifications as requiring no backprojection may be too accurate. More precisely, at higher levels $i > 0$ regions covered by an MSSM sample only align with texel boundaries at level 0 but not necessarily with the boundaries of the texture space support of the micro-occluders later constructed at levels $j > 0$ during visibility determination. As a consequence those micro-occluders overlapping the search area border get ignored which only become relevant micro-occluders because of the contribution of samples outside the search area during minimum aggregation. If this "over-precision" ultimately results in classifying

Figure 8: (Integer) shadow map levels used to meet a budget $b_{\text{mo}} = 16$ when employing (from left to right) the HSM, the MSSM without and with query region adaptation, and the YSM for search area refinement. Far right: soft shadows obtained with the YSM.

a pixel as fully lit, discontinuity artifacts can appear at the interface between penumbra and completely lit regions (see Fig. 7). To alleviate this issue, we determine the level j from the current search area and budget b_{mo}, and grow the area before querying its depth range to completely include the support regions of all partially covered level-j micro-occluders. Note that this effectively reduces the resolution of the higher MSSM levels.

Acceleration structure	Size	Construction time	Memory footprint
HSM	1024^2	0.26 ms	10.67 MB
MSSM	1024^2	3.92 ms	88.00 MB
YSM	$1024^2/256^2$	0.46 ms	14.67 MB
HSM	2048^2	0.87 ms	42.67 MB
MSSM	2048^2	18.65 ms	384.00 MB
YSM	$2048^2/256^2$	1.07 ms	46.67 MB

Table 1: Cost of acceleration structures on a GeForce 8800 GTX. Note that the YSM figures are for a non-truncated HSM part.

Y shadow map Considering required resources, quality of results and the fact that a full-resolution MSSM doesn't make much sense for higher levels if the budget b_{mo} is limited, we suggest employing a hybrid between HSM and MSSM. We obtain our so-called *Y shadow map* (YSM) by combining the first n levels of the HSM with an MSSM built from level $n-1$ of the HSM. (MSSM level 0 and HSM level $n-1$ are identical and hence MSSM level 0 is not explicitly stored.) The YSM is stored distributed over a 2D texture with mip-map chain for the HSM part and a 2D texture array for the MSSM part.

In practice, choosing a resolution of 256^2 for the MSSM part provides a good trade-off between additional construction cost and memory footprint compared to the HSM on the one hand and potentially less tight search areas compared to a full-resolution MSSM on the other hand (cf. Table 1 and Fig. 8). In our experience, the sometimes slightly increased number of pixels requiring micro-occluder processing when using the YSM in lieu of the MSSM is usually more than compensated for by the lower construction times.

While it is possible to truncate the HSM pyramid at level n where the MSSM begins, in our implementation we actually construct a full HSM when using micropatches or microquads. Although this is of no use for search area pruning, it allows us to always use the HSM for micro-occluder construction during visibility determination, improving texture fetch locality and avoiding branching.

Finally, we like to acknowledge that the basic idea of a hybrid approach has already been mentioned before [21], but note that to the best of our knowledge it has never been pursued or further investigated in practice.

7 Visibility Interpolation for Multisample Support

Overall visual quality of rendered scenes can often profit from multisample anti-aliasing (MSAA), and since multisampled rendering usually only incurs a negligible overhead with the latest graphics hardware, in many applications it is used by default. Accordingly, it is desirable to be able to provide soft shadows in such multisampled setting with a SSM algorithm.

In principle, this could be done by putting both shading and all SSM computations into a single big shader that gets executed when rendering the scene (after a depth-only render pass to avoid SSM computations for finally overdrawn fragments). However, apart from precluding the use of intermediate values from neighboring pixels as done in the scheme introduced in Subsection 4.2, this approach also leads to a low utilization of GPU resources because of low effective concurrency, especially in case of highly-tessellated scenes, resulting in significantly increased frame times.

Since Direct3D 10 [6] enables accessing individual multisamples, deferred shading approaches become possible as the final multisample resolve can be done in a fragment shader. Hence we also could use a multisampled visibility buffer as well as multisampled parameter buffers if required and fill them by rendering the whole geometry each time, which can have severe performance implications as mentioned above, however. Alternatively, we could use normal textures of double the size (in case of 4× MSAA) for visibility and the like, and draw a quad to trigger the computations. But since this implies doing four times the work compared to the single-sampled case, in order to roughly maintain the frame rate, we would have to resort to the next coarser level as this leads to about four times less micro-occluders to process per pixel.

Single visibility sample approach Notwithstanding the quadrupled memory footprint for the intermediate textures, all these approaches incur a price too high to pay for MSAA support. We hence reckon that it is better to accept minor imprecisions at the sub-pixel level that don't introduce noticeable artifacts if this allows to maintain soft shadow quality without substantially increasing frame time. Motivated by this reasoning, our approach first picks for each pixel a single multisample value s_z from the multisampled camera depth buffer and stores it in an ordinary texture. We then run our SSM algorithm as in the single-sample case and apply the resulting visibility buffer to the multisampled color buffer with a custom resolve, interpolating visibility for uncovered multisample values if necessary.

Note that a pixel's multisamples are only different if more than one primitive partially covers the pixel. Except in case of centroid sampling, all multisamples are taken at the same fractional screen-space position, extrapolating the underlying primitive if necessary. As a consequence, if a pixel is covered by multiple primitives, world position and hence camera depth of the multisamples differ unless the primitives are co-planar (and even then limited numerical precision of the rasterizer can lead to differences). Therefore, it is reasonable to assume that a pixel's multisamples are identical if their camera depth difference is below a certain threshold τ. In conclusion, usually the vast majority of pixels require only a single visibility determination.

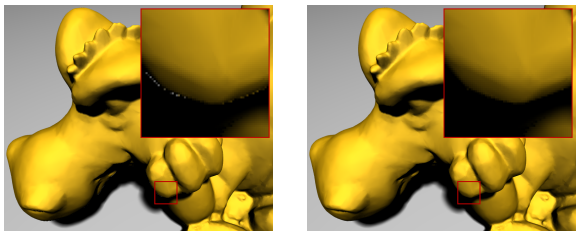

Figure 9: Combining multisampled rendering and single-sampled visibility determination without (left) and with (right) considering neighboring samples.

For the remaining pixels, where not all multisamples are equivalent to s_z within τ, we additionally look at neighboring pixels for each multisample which deviates from s_z by more than τ and take that visibility value whose corresponding camera depth value s_z is closest to the multisample's one.

We obtained good results (cf. Fig. 9) when considering a four-neighborhood (direct neighbors in x and y direction) for visibility interpolation and always picking the first multisample for the single-sampled SSM computation. However, note that for extremely fine structures like twigs with sub-pixel diameter more sophisticated methods for selecting the representative multisample may be worth exploring.

8 Results

We tested our algorithms on several scenes and with various settings. All reported results were obtained with an NVIDIA GeForce 8800 GTX at a viewport of size 1024×768, and utilized a YSM of size $1024^2/256^2$ for search area pruning.

Figure 10: Dinosaur under palm tree: (a) our scheme with $b_{\text{mo}} = 10$, (b) SSM using only level $i = 0$, and (c) reference solution (obtained by averaging the results of 1024 shadow maps).

Smooth quality variation Remember that the objective is to allow for spatially varying quality degradation without introducing transition artifacts, and not to improve on the highest quality possible with previous SSM schemes. A lower quality usually manifests itself in larger shadow regions as well as decreasing smoothness and detail in the shadow shape, but remains plausible (cf. Fig. 10). As demonstrated in Fig. 1, our approach successfully helps avoiding artifacts due to shadow map level transitions inherent to SSM with limited budget b_{mo}. The concept of a budget map additionally allows locally adapting soft shadow quality while retaining smooth transitions. For instance, we can selectively lower soft shadow quality at screen borders (cf. Fig. 1 d) which may be considered visually less important, or in regions of high texture masking (cf. Fig. 11) where a high soft shadow quality wouldn't be noticeable anyway.

Concerning performance, we first note again that blending is required to avoid transition artifacts unless all pixels use the same shadow map level i for micro-occluder construction. Since completely resorting to a coarser common shadow map level is usually not a viable option as objectionable artifacts would appear where occluders and receivers touch, we compare against using the finest level $i = 0$ throughout the screen (i.e. $b_{\text{mo}} = \infty$). As listed in Table 2, selectively lowering the quality and performing blending is

Scene	b_{mo}	fps	pixels	avg	stdev
Fig. 1 b	5	82.6	15.6%	33.3	13.7
Fig. 1 d	5–12	35.8	13.0%	164	138
Fig. 1 c	12	27.1	11.7%	480	313
Fig. 1	∞	23.9	11.0%	548	351
Fig. 8	9	46.7	9.2%	158	80.2
Fig. 8	∞	15.0	8.0%	1372	1003
Fig. 10 a	10	24.8	12.7%	194	84.8
Fig. 10 b	∞	7.1	11.2%	3632	2166

Table 2: Performance quantities for various setups: frame rate, percentage of pixels performing backprojection of micro-occluders, and mean and standard deviation of the number of processed micro-occluders for these pixels. In case of an unconstrained budget (i.e. $b_{\text{mo}} = \infty$) no LOQ map smoothing or alpha blending is performed.

indeed faster than always using the finest level. This is mainly due to having significantly less micro-occluders to process per pixel on average. On the other hand, note that the percentage of pixels for which micro-occluders actually need to be backprojected increases with decreasing budget b_{mo} because during search area pruning the query regions are adapted according to b_{mo} to avoid "over-precision" artifacts, causing additional pixels (close to penumbrae that would be obtained for $i = 0$) to perform actual backprojections.

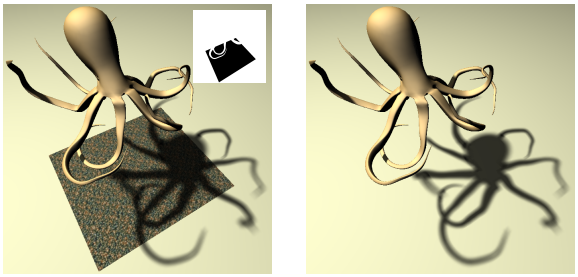

Figure 11: Our scheme allows to selectively reduce the budget b_{mo} in regions of high texture masking (left: from 12 to 5, see budget map inset) to increase performance (50 fps) compared to retaining a uniform budget (right: 32 fps).

Our scheme features three sources of overhead compared to normal SSM. First, due to distributing the calculations among several shaders, texture reads and writes to the search area buffer, LOQ map and budget map are introduced. However, we didn't observe any measurable performance penalty, probably because the effective parallelism of fragment shader executions and cache utilization are improved by this reorganization. Second, the LOQ map is smoothed in a separate pass. We use a filter with support 11×11, which provides a good trade-off between speed and smoothing capability. Our measurements show that the smoothing step takes less than 2 ms. Finally, if the smoothly varying shadow map level ℓ is fractional, we blend the visibility results for levels $\lfloor \ell \rfloor$ (where at most $(2b_{\text{mo}})^2$ micro-occluders get processed) and $\lceil \ell \rceil$ (at most $(b_{\text{mo}})^2$ micro-occluders), and hence process more micro-occluders per pixel than with normal SSM where only level $\lfloor \ell \rfloor$ is considered. Measurements across many scenes and viewpoints suggest that in practice about 30% more micro-occluders are considered on average. However, in return for these overheads, spatial transition artifacts are successfully alleviated, enabling the practical use of selectively lower soft shadow quality to reduce frame time.

Microrects Microrects differ from and improve on micropatches at coarser levels $i > 0$. As shown in Fig. 12, in such lower quality settings, they yield superior results compared to micropatches, thanks to more accurately approximating the underlying occluder geometry. We also observe that when our blending

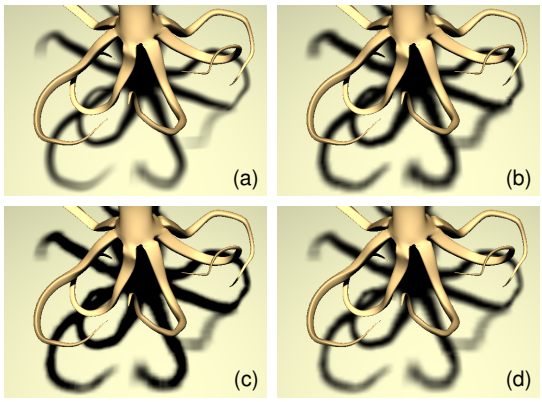

Figure 12: Soft shadows obtained with (a) 1024 shadow maps, (b) coarse micropatches and occlusion bitmasks, (c) coarse microrects with area accumulation and (d) with occlusion bitmasks.

scheme is not used, the resulting transition artifacts are often less pronounced than in case of micropatches (cf. Fig. 13).

The construction of the two microrect textures adds a minor overhead of 1.89 ms for a 1024^2-sized shadow map. Unfortunately, mainly because of having to use occlusion bitmasks due to many overlaps, the shader register count is increased slightly compared to simple area accumulation of micropatches, causing a slow-down of up to 72%, with the utilization of (jittered 16×16) occlusion bitmasks already accounting for a slow-down of up to almost 50%. Further tests on an AMD Radeon HD 2900 XT however showed significantly lower performance drops, with overall slow-downs being less than 35%, rendering microrects still interesting for practical use already now and a promising candidate for upcoming hardware with larger register files. We also note that since both the probability and the average size of micro-occluder overlaps increase at coarser levels, using occlusion bitmasks is often reasonable, anyway.

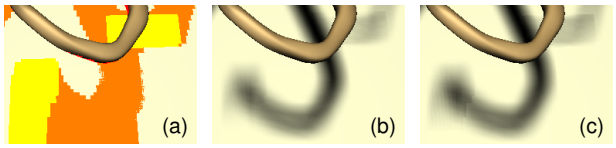

Figure 13: When multiple (integer) shadow map levels are used across penumbra pixels (a; color key as in Fig. 8), microrects (b) often cause less noticeable transition artifacts than micropatches (c).

Multisample support Our visibility interpolation approach enables soft shadows for multisampled rendering at low extra cost compared to single-sampled setups. Measurements over a range of scenes of different complexity suggest that with our scheme multisampling introduces an overhead of less than 9%. We note that multisampling itself, i.e. without applying our technique, already accounts for an impact of about 5% on the frame time.

9 Conclusion and Future Work

We have dealt with various aspects concerning soft shadow quality of SSM. First, we have investigated options to vary soft shadow quality in general and suggested a practical scheme for SSM that enables smoothly changing quality across screen and, in particular, allows locally adapting quality according to a budget map.

Moreover, we introduced microrects which offer better occluder approximations at coarser shadow map levels, hence enabling a higher visual quality for a given micro-occluder budget. In addition, a hybrid acceleration structure has been presented which combines extensive search area pruning with low resource requirements, positively affecting overall quality as often finer shadow map levels can be used for visibility computations. Finally, we have proposed an interpolation scheme that allows retaining the soft shadow quality achievable in single-sample rendering within multisampled settings at a comparably negligible cost.

In future work, it would be interesting to investigate further aspects affecting quality like biasing and its interplay with using varying shadow map levels.

Acknowledgements

This work was funded by the European Union within the CROSSMOD project (EU IST-014891-2).

References

[1] J. Arvo, M. Hirvikorpi, and J. Tyystjärvi. Approximate soft shadows with an image-space flood-fill algorithm. *Computer Graphics Forum*, 23(3):271–280, 2004.

[2] U. Assarsson and T. Akenine-Möller. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Transactions on Graphics*, 22(3):511–520, 2003.

[3] B. Aszódi and L. Szirmay-Kalos. Real-time soft shadows with shadow accumulation. In *Eurographics 2006 Short Presentations*, pages 53–56, 2006.

[4] L. Atty, N. Holzschuch, M. Lapierre, J.-M. Hasenfratz, C. Hansen, and F. X. Sillion. Soft shadow maps: Efficient sampling of light source visibility. *Computer Graphics Forum*, 25(4):725–741, 2006.

[5] L. Bavoil and C. T. Silva. Real-time soft shadows with cone culling. In *ACM SIGGRAPH 2006 Sketches and Applications*, 2006.

[6] D. Blythe. The Direct3D 10 system. *ACM Transactions on Graphics*, 25(3):724–734, 2006.

[7] X.-H. Cai, Y.-T. Jia, X. Wang, S.-M. Hu, and R. R. Martin. Rendering soft shadows using multilayered shadow fins. *Computer Graphics Forum*, 25(1):15–28, 2006.

[8] E. Eisemann and X. Décoret. Visibility sampling on GPU and applications. *Computer Graphics Forum*, 26(3):535–544, 2007.

[9] R. Fernando. Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches and Applications*, 2005.

[10] X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. *ACM Transactions on Graphics*, 21(3):355–361, 2002.

[11] G. Guennebaud, L. Barthe, and M. Paulin. Real-time soft shadow mapping by backprojection. In *Rendering Techniques 2006*, pages 227–234, 2006.

[12] G. Guennebaud, L. Barthe, and M. Paulin. High-quality adaptive soft shadow mapping. *Computer Graphics Forum*, 26(3):525–533, 2007.

[13] J.-M. Hasenfratz, M. Lapierre, N. Holzschuch, and F. Sillion. A survey of real-time soft shadows algorithms. *Computer Graphics Forum*, 22(4):753–774, 2003.

[14] H. Hoppe. View-dependent refinement of progressive meshes. In *Proceedings of ACM SIGGRAPH 97*, pages 189–198, 1997.

[15] A. Lauritzen. Summed-area variance shadow maps. In H. Nguyen, editor, *GPU Gems 3*, pages 157–182. Addison Wesley, 2007.

[16] D. Luebke and C. Erikson. View-dependent simplification of arbitrary polygonal environments. In *Proceedings of ACM SIGGRAPH 97*, pages 199–208, 1997.

[17] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner. *Level of Detail for 3D Graphics*. Morgan Kaufmann, 2002.

[18] T. Q. Pham and L. J. van Vliet. Separable bilateral filtering for fast video preprocessing. In *Proceedings of IEEE ICME 2005*, pages 454–457, 2005.

[19] W. T. Reeves, D. H. Salesin, and R. L. Cook. Rendering antialiased shadows with depth maps. In *Computer Graphics (Proceedings of ACM SIGGRAPH 87)*, volume 21, pages 283–291, 1987.

[20] D. Scherzer, S. Jeschke, and M. Wimmer. Pixel-correct shadow maps with temporal reprojection and shadow test confidence. In *Rendering Techniques 2007*, pages 45–50, 2007.

[21] M. Schwarz and M. Stamminger. Bitmask soft shadows. *Computer Graphics Forum*, 26(3):515–524, 2007.