EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2016)
C. E. Catalano and L. De Luca (Editors)

Multi-View Ambient Occlusion for Enhancing Visualization of Raw Scanning Data

Manuele Sabbadin[1,2], Gianpaolo Palma[1], Paolo Cignoni[1], Roberto Scopigno[1]

[1]Visual Computing Lab - ISTI - CNR
[2]Department of Computer Science - University of Pisa

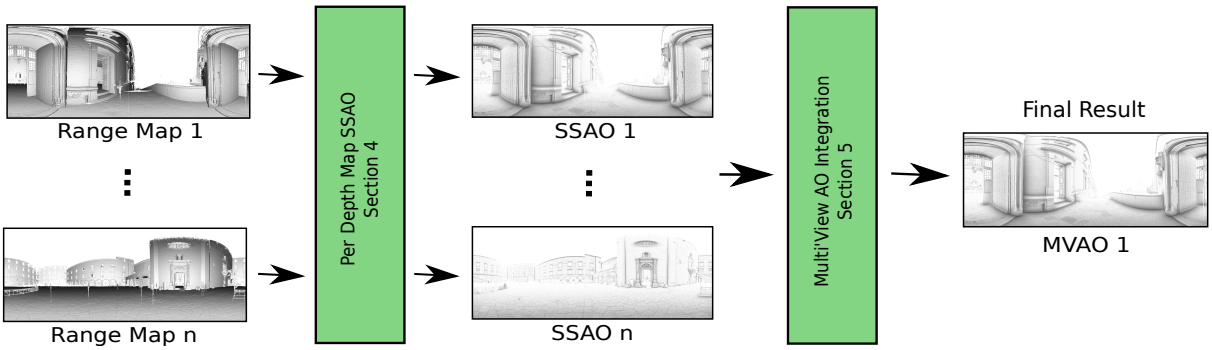

Figure 1: *Algorithm overview. Given a set of input scans arranged as range maps the algorithm computes the Screen Space Ambient Occlusion on each of them, using the method presented in Section 4. The Multi-View AO integration step (Section 5) merges the information of these maps to obtain a more robust estimation in the regions occluded by objects not visible in a single scan and in the regions with different local density among the scans.*

Abstract

The correct understanding of the 3D shape is a crucial aspect to improve the 3D scanning process, especially in order to perform high quality and as complete as possible 3D acquisitions on the field. The paper proposes a new technique to enhance the visualization of raw scanning data based on the definition in device space of a Multi-View Ambient Occlusion (MVAO). The approach allows improving the comprehension of the 3D shape of the input geometry and, requiring almost no preprocessing, it can be directly applied to raw captured point clouds. The algorithm has been tested on different datasets: high resolution Time-of-Flight scans and streams of low quality range maps from a depth camera. The results enhance the details perception in the 3D geometry using the multi-view information to make more robust the ambient occlusion estimation.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms I.3.3 [Computer Graphics]: Picture/Image Generation—Digitizing and scanning

1. Introduction

Ambient Occlusion (AO) [Lan02] and Obscurance [ZIK98] are two well known techniques that are used to generate effective approximations of global illumination effects in rendering. The impact of these lighting cues in improving the readability and understandability of 3D shapes is well known [TCM06] [LB99], overcoming the effectiveness of a simple Lambertian shading (Figure 2). However the use of these visual enhancing techniques in the management of raw scanning and acquisition data remained very limited, in spite of the fact that in this context the comprehension of 3D shapes and their relations is critical for performing a complete and high quality acquisition. One of the main issues that prevents the use of AO in these context is the efficient computation of the global illumination effects of the scene. Since this computation is hard and complex, several real-time methods have been proposed to approximate these global effects by using simplifying assumptions. Until now the research attention was focused on new methods to precompute the AO value or to approximate it during the final ren-

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

dering using screen-space approaches. Moreover the intrinsically incomplete, noisy and fragmentary nature of scanned data makes difficult the direct application of the existing approaches: sparse point clouds offer small support for a direct application of ambient occlusion definitions.

This paper proposes a new algorithm that, taking advantage of a multi-view 3D acquisition with multiple range maps, is able to quickly compute a good approximation of the AO value for the final scanned 3D model (Figure 2). Our contribution is a framework that is able to integrate robustly the AO value computed for each single range map with a fast screen-space technique, starting from a 3D acquisition with or without temporal coherence. In the first case we propose an algorithm that process independently each range map after the acquisition and then integrates the AO value between the different scans, while in the second case we use the temporal coherence of the acquisition device (for example a RGB-Depth sensor) to estimate the evolving AO value in real time during the acquisition. The integration of the AO data is performed with a new robust weighting function that solves the usual view-dependent artifacts of the screen-space techniques, mainly due to the absence of occluders not visible in a single view, and avoids abrupt changes of AO value near to regions not acquired by all the scans and with large differences of local density (see Figure 5). On the other hand the use of a per-scan Screen Space AO, computed directly on the relative depth map that produced the point cloud, permits to work directly with the raw point clouds acquired by the scanner without the need of a 3D triangulation (see Figure 2). We finally show the results on a big dataset of a portion of a large and complex building, the Maschio Angioino in Naple, composed by 50 high resolution Time-of-Flight range scans.

2. Background and Related Work

Ambient Occlusion and Obscurance are similar algorithms able to efficiently simulate global illumination effects, like the diffuse indirect illumination. Their aim is to give a more realistic look to a 3D model diffusely illuminated, darkening its appearance in the regions that are more occluded. The main difference is that the Obscurance uses a more physical representation of the scene illumination, taking into account the distance from the occluder, while the ambient occlusion is limited to the openness of the point.

The Obscurance was presented in [ZIK98] to be used in videogame environments, where the value are precomputed and stored in a texture map attached to the 3D model. The Obscurance value of a point $\mathbf{x}$ is defined as:

$$O(\mathbf{x},\vec{\mathbf{n}}) = \frac{1}{\pi}\int_{\vec{\omega}\in\Omega} \rho(d(\mathbf{x},\vec{\omega}))(\vec{\mathbf{n}}\cdot\vec{\omega})\mathrm{d}\vec{\omega} \qquad (1)$$

where $\vec{\mathbf{n}}$ is the normal of the point $\mathbf{x}$, $\vec{\omega}$ is a direction in the hemisphere over $\mathbf{x}$, $d(\mathbf{x},\vec{\omega})$ is the distance of $\mathbf{x}$ to the first intersected point in direction $\vec{\omega}$, ρ is a monotone increasing function of the distance d with values in $[0,1]$ giving the magnitude of incoming ambient light from direction $\vec{\omega}$, $1/\pi$ is the normalization factor. The amount of indirect illumination computed depends on the maximum distance used in the function ρ beyond which the value is always equals to 1.

The Ambient Occlusion technique was presented in [Lan02] for the production rendering in the movies industries. It defines the percentage of occlusion of a point $\mathbf{x}$ on a surface as:

$$AO(\mathbf{x},\vec{\mathbf{n}}) = \frac{1}{\pi}\int_{\vec{\omega}\in\Omega} V(\mathbf{x},\vec{\omega})(\vec{\mathbf{n}}\cdot\vec{\omega})\mathrm{d}\hat{\omega} \qquad (2)$$

where V is the visibility function along the direction $\vec{\omega}$, which is 0 if occluded and 1 otherwise.

The algorithms for the computation of the Ambient Occlusion and Obscurance can be classified in two categories: object-space and screen-space methods. An example of object-space method was proposed in [McG10] using the computation on GPU of an occlusion volume by scattering of the AO contributions of each primitive in the surrounding space. The scattering is limited by a cutoff distance that affects the performance. For a complete review of the object-space approaches see [MFS09].

Screen-space techniques compute occlusion using the information in the G-buffer (depth + normal). This approach was introduced in [Mit07] by sampling a sphere centered at the shaded fragment and testing if each sample is occluded by the geometry using the depth buffer info. Shanmugam et al. [SA07] propose a multipass algorithm that separates the ambient occlusion in two terms: high frequency due to high surface details and low frequency due to distant occluders. Horizon-Based Ambient Occlusion (HBAO) [BSD08] uses the concept of horizon mapping to accumulate the occlusion from a set of azimuthal directions, finding the largest occluder in each direction by ray marching. Volumetric Obscurance (VO) [LS10] formulates the problem as a 3D volumetric integral solved using line samples. It measures the exact distance between the hemisphere around the point and the actual surface at a number of samples uniformly selected on a disk around the shaded point. Concurrently, another volumetric approach based only on screen space depth values was proposed in [SKUB*10]. Alchemy Ambient Obscurance [MOBH11] combines the sampling scheme of VO with the projected solid-angle occlusion of HBAO. This algorithm is extended in [MML12] with a set of architecture-aware performance and integration improvements. Line-sweep AO [Tim13] computes occlusion along a set of azimuthal directions by performing line sweeps across the depth buffer in each direction, reducing the complexity of the algorithm. Hendrickx et al. [HSEE15] propose an efficient statistical model to evaluate the volumetric occlusion factor in screen-space using a single sample. Overestimation and halos are reduced by an adaptive layering of the visible geometry. Depth-peeled screen space layers are used in [BS09] to improve the quality of the occlusion estimation. The method, using an enlarged field of view, calculates the AO for each depth layer and keeps the highest occlusion. Ritschel et al. [RGS09] show how the screen-space occlusion methods can be used to compute other rendering effects than just occlusion. To improve the rendering quality when a blocker or source of indirect light is not visible in the depth buffer, they propose to use additional cameras with different positions. Vardis et al. [VPG13] propose a generic per-fragment view weighting scheme for evaluating screen-space occlusion using multiple, arbitrary views, such as the readily available shadow maps. Additionally, it exploits the resulting weights to perform adaptive sampling based on the importance of each view in order to reduce

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

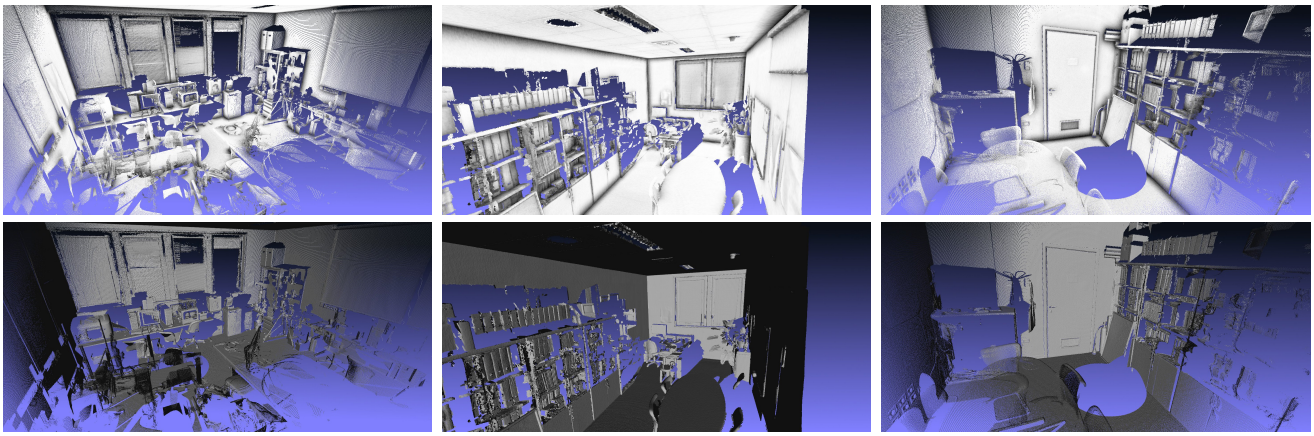

Figure 2: *Multi View Ambient Occlusion (MVAO) results mapped on the original point clouds: (Top) rendering of the point cloud with a mapping of the MVAO on the vertex color and without shading; (Bottom) rendering of the point cloud with a per vertex Lambertian shading.*

the total number of samples. Our algorithm extends this method in order to work with an arbitrary number of views, like for example the output of a moving RGB-Depth camera that produces a continuous streams of depth maps, and to use a more robust per-view weighting function.

3. Overview

The paper proposes a technique to visualize raw point clouds coming from 3D scanning in a more effective way. The proposed approach is based on a robust and fast technique to compute the Ambient Occlusion contribution exploiting the scanning information. A point cloud coming from a 3D acquisition is usually made by multiple scans of the subject (object or environment), taken from different point of views, to cover the whole surface. Each scan is organized as a depth map, an image containing in each pixel the depth of the surface from the scanner position. To guarantee the registration of the scans in the same coordinate system, so that they can be integrated into a single as complete as possible 3D model, the set of scans shows enough overlaps that can be used for the robust integration of the surface info computed for each scan independently. Our approach takes advantage of both the image-based nature of the point cloud and the redundancy coming from the multiple acquisitions to define in a robust way the AO computation of a point cloud. In the specific, starting from a set of scans with the relative alignment matrix to bring them in the same coordinate system, the algorithm completes two steps:

1. the computation of the Screen Space AO (SSAO) for each scan/depth map independently, using the scanner position;
2. the integration of the per-view AO maps, finding the AO values corresponding to the same point in all the scans and merging them with a robust weighting function.

Since each map contains data not available in the other views, integrating the AO from multiple range maps, acquired from different positions, allows solving the view-dependent inconsistencies of the SSAO algorithms due to the lack of geometric information in each single depth map (for example missing occluders). The final result is a more consistent and coherent global AO value. The algorithm assumes static scene during the 3D acquisition. Moreover the main advantage of the computation of the SSAO on the depth maps is the possibility to work directly on the raw scanned data without any other intermediate processing.

In the following sections we describe the SSAO algorithm used for each scan (Section 4) and two different integration schemes of the per-view AO maps suitable for opposite acquisition case: acquisition without temporal coherence (Section 5) where we have few high resolution scans of the same object taken with an high end scanner (for example laser triangulation or LIDAR scanner); acquisition with temporal coherence (Section 6) where we have a real-time stream of low resolution depth maps acquired with a depth camera (for example the Microsoft Kinect). To note that the proposed algorithm can use any state of the art SSAO approach instead of the presented one in the Section 4. Finally in the Section 7 we discuss the obtained results, showing also the application of the proposed algorithm to a 3D dataset of a Cultural Heritage building, the interior of the Maschio Angioino in Naple, characterized by a complex architectural structure.

4. Per Depth Map SSAO

For the computation of the SSAO we implement a simple algorithm based on the Equation 1. Given the G-buffer (position + normal) of the scan, obtained by its rendering from the scanner position using the scanner camera model (for example the equirectangular projection camera for a LIDAR scanner, or the perspective camera model for a laser triangulation scanner), we compute the AO value for each valid point with buffer coordinates (x,y), 3D position $\mathbf{p}$ and normal $\vec{\mathbf{n}}$ by approximating the Equation 1 with a Monte Carlo sampling of a screen space disk around (x,y) such that:

$$AO(x,y) = \frac{1}{\#S} \sum_{\forall \mathbf{p_i} \in S} \rho(\mathbf{p},\mathbf{p}_i)\, w(\mathbf{p},\vec{\mathbf{n}},\mathbf{p}_i) \qquad (3)$$

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

where:

$$\rho(\mathbf{p},\mathbf{p}_i) = \text{smoothstep}(0, t_{max}, \|\mathbf{p}_i - \mathbf{p}\|) \quad (4)$$

$$w(\mathbf{p},\vec{\mathbf{n}},\mathbf{p}_i) = \left(1 - \max\left(\vec{\mathbf{n}} \cdot \frac{\mathbf{p}_i - \mathbf{p}}{\|\mathbf{p}_i - \mathbf{p}\|}, 0\right)\right) \quad (5)$$

The set S contains the 3D positions of the samples generated in the screen-space disk around (x,y). The size of the disk is adaptive with respect to the depth of the point $\mathbf{p}$ from the camera position to guarantee a sampling sphere of the same area for all the points: the deeper is the point the smaller is the screen-space disk radius. The adaptive radius is very important for the scans where there is big differences of depths, in order to compute AO values with the same scale of details for all the points. In practice, we compute this radius by projection in camera space of a sphere with a user selected radius r_s and center in $\mathbf{p}$. To speed-up the algorithm we do not compute the accurate projection of this sphere in image space using the scanner intrinsics parameters because it requires the estimation of the matrix form of a generic implicit conic. Instead we assign to the points with the same depth the same disk radius, independently from the image coordinates (x,y), by moving the sphere on the view axis of the camera keeping the original depth of the point. In this way its screen-space projection is always a circle and the estimation of its radius is simpler. Inside the computed disk we generate N random samples (for our experiments 32) with a uniform Monte Carlo sampling and for each sample we retrieve the relative 3D position $\mathbf{p}_i$.

In Equation 3, ρ is a monotone increasing function of the distance between the point $\mathbf{p}$ and the sample $\mathbf{p}_i$, implemented with a cubic Hermite interpolation using a maximum distance t_{max} selected by the user. The farther is the sample $\mathbf{p}_i$ the less is its contribution in the occlusion of $\mathbf{p}$. The function w implements the visibility attenuation factor: the smaller is the visibility angle between the points, the higher is the occlusion of the sample $\mathbf{p}_i$.

The two user parameters r_s and t_{max} determine the scale at which to compute the occlusions. The higher is r_s the softer and more blurred are the AO halos, while on the contrary the higher is t_{max} the darker and sharper are the AO halos (Figure 3). The area of the geometry affected by the AO increases when the two parameters grow. The SSAO values are finally blurred with a Gaussian 5×5 smoothing to reduce the sampling noise.

5. Multi-View AO Integration

In this section we describe a new algorithm to merge together the AO maps computed with the SSAO approach presented in the Section 4 for a set of scans $\{v_1, \cdots, v_n\}$ acquired without temporal coherence. The main idea is similar to the method presented in [VPG13] where the AO contribution of different views generated during the rendering of the 3D model is weighted with a function of normal and depth of the samples. We extend this method with a more robust weighting scheme.

The proposed method requires that the scans are aligned in a common reference system and for each scan v_i we know the alignment matrix $\mathbf{M}_i$ to move it in this system. The alignment can be precomputed automatically during or after the acquisition. In the former case the use of special markers allows the scanner to compute the right alignment matrix while in the later case we can compute a rough alignment of the scans [PCGS15] followed by a further refinement with the ICP algorithm [BM92]. The algorithm requires also a radius information for each point in the scans estimated as $\text{rad}(\mathbf{p}) = 2\sqrt{d_k^2/k}$ where d_k is the radius of the smaller sphere centered in $\mathbf{p}$ that contains the k-nearest points ($k = 32$). If the input scans are not equipped with normals we estimate them using the method in [HDD*92] with 16 nearest points.

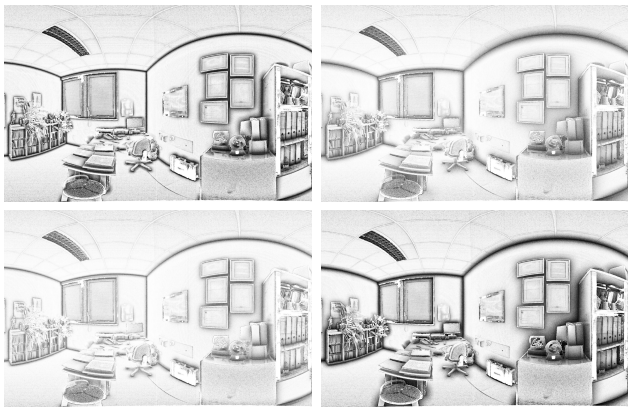

Figure 3: *SSAO results with different values for r_s and t_{max}: (Top-left) $r_s = 10cm$ and $t_{max} = 40cm$; (Top-right) $r_s = 40cm$ and $t_{max} = 40cm$; (Bottom-left) $r_s = 20cm$ and $t_{max} = 20cm$; (Bottom-right) $r_s = 20cm$ and $t_{max} = 80cm$.*

The algorithm computes the Multi View Ambient Occlusion (MVAO) value for each point in each scan. The first step is to retrieve the corresponding points from the other scans using the alignment matrix. For each point $\mathbf{p}_i$ in the scan v_i we compute the set L:

$$L = \{\mathbf{p}_{ij} \mid \mathbf{p}_{ij} = \text{proj}\,(\mathbf{M}_j^{-1}\mathbf{M}_i\mathbf{p}_i\,,\, v_j)\} \quad (6)$$

where $\mathbf{M}_j^{-1}\mathbf{M}_i\mathbf{p}_i$ transforms the point $\mathbf{p}_i$ in the local space of the scan v_j and the function *proj* projects the point in the G-buffer of the scan v_j and returns the real acquired 3D position. For the final integration the algorithm selects a subset $L_2 \subseteq L$ taking only the points near to $\mathbf{p}_i$ and visible from the scanner used in the scan v_i:

$$L_2 = \{\mathbf{p}_{ij} \in L \mid \left\|\mathbf{M}_j^{-1}\mathbf{M}_i\mathbf{p}_i - \mathbf{p}_{ij}\right\| < 2t \wedge (\mathbf{b} - \mathbf{p_{ij}}) \cdot \vec{\mathbf{n}}_{ij} > 0 \} \quad (7)$$

where t is the maximum radius between $\mathbf{p}_i$ and $\mathbf{p}_{ij}$, $\vec{\mathbf{n}}_{ij}$ is the normal of $\mathbf{p}_{ij}$ and $\mathbf{b}$ is the position of the scanner used in the scan v_i and transformed in the local coordinate system of the scan v_j. The final MVAO value for the point $\mathbf{p}_i$ is equal to the weighted average of the per depth map SSAO value of the points in L_2:

$$MVAO(\mathbf{p}_i) = \frac{\sum_{\mathbf{p}_{ij} \in L_2} SSAO_{v_j}(\mathbf{p}_{ij})\, w_{v_j}(\mathbf{p}_{ij}, \mathbf{p}_i)}{\sum_{\mathbf{p}_{ij} \in L_2} w_{v_j}(\mathbf{p}_{ij}, \mathbf{p}_i)} \quad (8)$$

where $SSAO_{v_j}(\mathbf{p}_{ij})$ returns the SSAO value of the point $\mathbf{p}_{ij}$ in the scan v_j and $w_{v_j}()$ returns the relative weight. The main contribution of our algorithm is the definition of a new robust weighting function

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

w that is the product of three factors:

$$w_{v_j}(\mathbf{p}_{ij}, \mathbf{p}_i) = w_d(\mathbf{p}_{ij})\, w_n(\mathbf{p}_{ij})\, w_b(\mathbf{p}_{ij}, \mathbf{p}_i) \tag{9}$$

where for each sample we have the distance weight w_d, the directional weight w_n and the border weight w_b.

The distance weight gives more influence to the samples that are acquired by closer scanners with more details due to the higher resolution. The function is defined as:

$$w_d(\mathbf{p}_{ij}) = \max\left(1 - \frac{\max(\|\mathbf{p}\| - d_{min}, 0)}{d_{max}}, 0.1\right) \tag{10}$$

where $\|\mathbf{p}\|$ is the depth from the scanner position and d_{max} and d_{min} are the maximum and minimum normalization depth thresholds. Using small d_{max} value reduce the effectiveness of the function w_d (Figure 4). A good selection for these thresholds are the maximum depth in all the depth maps and the minimum scanning distance of the acquisition device.

Figure 4: *Multi View Ambient Occlusion with different maximum depth threshold d_{max} in Equation 10: (Left) $d_{max} = 2m$; (Right) $d_{max} = 6m$.*

The directional weight gives more influence to the samples that are acquired from a more orthogonal view direction. The function is defined as:

$$w_n(\mathbf{p}_{ij}) = \max\left(\vec{\mathbf{n}} \cdot \hat{\vec{l}}, 0\right) \tag{11}$$

where $\vec{\mathbf{n}}$ is the normal of the sample and $\hat{\vec{l}}$ is the direction from $\mathbf{p}$ to the scanner position in the scan s_j.

Finally the border weight w_b is used to make more smooth the transaction between areas with different coverage percentage among the scans, for example close points in the depth map that have a different number of samples in the set L_2. The correct management of this kind of coverage discontinuities is very important when a low number of input scans is available (Figure 5). This weight is computed as the nearest distance d_b in pixel from a depth discontinuity in the depth map of the scan. In the specific, we extract the edges from the depth map using the image Laplacian operator, we detect the most valuable borders using the 0.95 percentile of the histogram of the edge maps and then we compute the distance field from these borders using a GPU jump flooding algorithm [RT06]. The resulting value is normalized in an adaptive way to take account of wide sampling density differences in the same

Figure 5: *Effect of the border weight w_b on the computation of the MVAO: (Top-left) image with the color encoding of the scan coverage for each point (yellow = one scan, light blue = two scans, pink = three scans); (Top-right) MVAO using our weighting function in Equation 9; (Bottom-left) MVAO using the weighting function proposed in [VPG13]; (Bottom-right) MVAO using our weighting function without the border weight w_b.*

point among the scans. The final border weight is computed as:

$$w_b(\mathbf{p}_{ij}, \mathbf{p}_i) = \min\left(\frac{d_b}{t_p \alpha(\mathbf{p}_{ij}, \mathbf{p}_i)}, 1\right) \tag{12}$$

where t_p is a threshold in pixel (in our case always $t_p = 64$) and $\alpha(\mathbf{p}_{ij}, \mathbf{p}_i) = \text{rad}(\mathbf{p}_i)/\text{rad}(\mathbf{p}_{ij})$ is the ratio of the radius of the points $\mathbf{p}_i$ and $\mathbf{p}_{ij}$. The idea of this adaptive normalization is to transform the pixel distance d_b in a weight with a uniform scale among the different scans, especially where the local density is very different.

Equation 8 can be generalized using N samples for each scan by taking for each point $\mathbf{p}_{ij} \in L_2$ the N pixel neighbors in the depth map (3×3, 5×5, ...). Each sample is weighted always with the same functions in Equation 9. Increasing the number of neighbors N produces smoother results (Figure 6).

6. Realtime Multi-View AO Integration

This section describes how to extend the algorithm in Section 5 for a realtime scenario, where a low resolution depth camera can generate a temporal and spatial continuous stream of depth maps. For our experiment we used as acquisition device the second version of the Microsoft Kinect. Since the number of scans acquired with this device grows constantly it would not be possible to integrate the AO value of all the range maps as seen before. Instead, we use a volumetric uniform grid, that we call MVAO-Grid, to represent the underlying scanning space and to accumulate and integrate the

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

Figure 6: *Effect of the number of samples N for each scans used in the Equation 8: (Left) 1 sample per scan; (Right)* 9×9 *samples per scan.*

SSAO values of the captured frames on the fly. Each cell of the grid contains two values: the accumulated SSAO and the relative accumulated weight. The values in the grid are used during the rendering of the current frame to make more realistic its visualization and at the end to save the final model with the integrated MVAO.

Algorithm 1 shows an overview of the different steps executed for each input depth map. In the algorithm we use the Microsoft KinectSDK methods to get the sensor data, to align the input depth map and to create the triangular mesh of the current acquired scene.

Algorithm 1 MVAO computation for each input frame

1: *frame* ← GETKINECTFRAME()
2: REMOVEOUTLIERS(*frame*)
3: M ← KINECTFUSIONINTEGRATION(*frame*)
4: *SSAOMap* ← GETSSAOANDWEIGHTS(*frame*)
5: UPDATEMVAOGRID(*SSAOMap*, M)
6: RENDER(*frame*)

After the acquisition of the current frame, the algorithm executes a cleaning on the depth map to remove the outliers near the borders of the scanned objects. This filter compares the depth of each point $\mathbf{p_i}$ to the depths of its 8 neighbors. If a neighbor presents a depth difference greater than a threshold $t = 2$cm we remove the point $\mathbf{p_i}$ from the depth map. Even if this method is very aggressive near the depth discontinuities, by removing more samples that needed, the removed good samples can be acquired again by moving the acquisition device in positions that allow a more orthogonal view of that geometry. The cleaned depth map is aligned with the previous acquired data using the KinectFusion approach [NIH*11]. This procedure merges the current depth in its internal volumetric representation and returns the alignment matrix $\mathbf{M}$. In the next stage the SSAO of the frame is computed as seen in Section 4. In addition for each point we calculate the weight of the SSAO value using a simplified version of the weighting function $w()$ in Equation 9 that does not use the border weight w_b, because its effects are approximated by the integration of the SSAO values in a common space given by the MVAO-Grid. Then we update the MVAO-Grid. Using the alignment matrix $\mathbf{M}$ we put the frame in the same reference system of the MVAO-Grid, we find the grid cells that contain the input points of the frame and we accumulate the SSAO value and weight of the points in that cells. The last step is the rendering of the frame with the MVAO values obtained by division of the accumulated values with the accumulated weights of the MVAO-Grid.

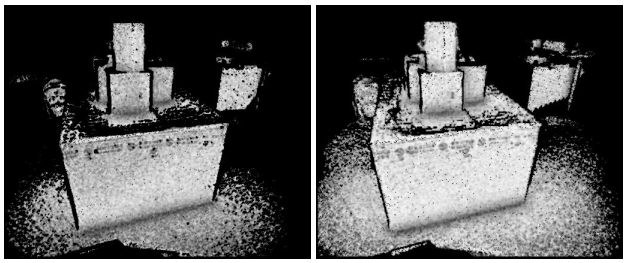

Figure 7: *SSAO computed on different inputs: (Left) raw depth map received by the Kinect; (Right) depth map generated by ray-casting of the volumetric distance function generated to merge the input data.*

To reduce the impact of the noise of the output depth maps on the SSAO computation, it is possible to compute the SSAO on a different depth map obtained by ray-casting of the volumetric truncated signed distance function created by the KinectFusion using the original data. The computation of this ray-casted depth map must be done between the line 3 and 4 of the Algorithm 1. The new depth map, that represents a rendering of the current fuse model, is less noisy showing better normals. Figure 7 shows a comparison of the SSAO computed for both the kind of depth maps.

7. Results

We implemented the algorithms in GPU using OpenGL and C++ and tested on an Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz, 32GB RAM and an NVidia GTX 980 graphics card. The multi view integration algorithm in Section 5 has been tested with two scanning datasets of two different rooms (a laboratory and an office) acquired with a Time-of-Flight laser scanner. Each dataset contains three point clouds acquired by different positions. Since the scanner do not save the acquired depth maps we transformed each point cloud in an equirectangular map of resolution $2h \times h$ ($h = \sqrt{\#v/3}$ with $\#v$ the maximum number of points in the clouds of the dataset) storing for each pixel the relative position and normal. Each point of the cloud is projected in the map using its spherical coordinates. The map resolution used for the input dataset is 3538×1769. The three scans are aligned semi-automatically using MeshLab [CCR08] saving the relative registration matrices. Figures 8 and 9 show the obtained results in these datasets comparing the SSAO with the MVAO. To enhance the visualization of the differences we applied to all the images a gamma correction with $\gamma = 4.0$. All the SSAO maps are generated with $r_s = 0.2$m and $t_{max} = 0.4$m. We can see how the MVAO algorithm robustly merges the SSAO values of all the scans improving the perception of some details in each scan. In particular, in the nearest regions to the scanner position, where the acquired data are more noisy, the MVAO computes a better and cleaner AO value, while in the farther areas it adds important details giving more weight to the scans that acquired these regions with more resolution. Moreover it adds the occlusions due to objects not visible in the scan. Figures

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

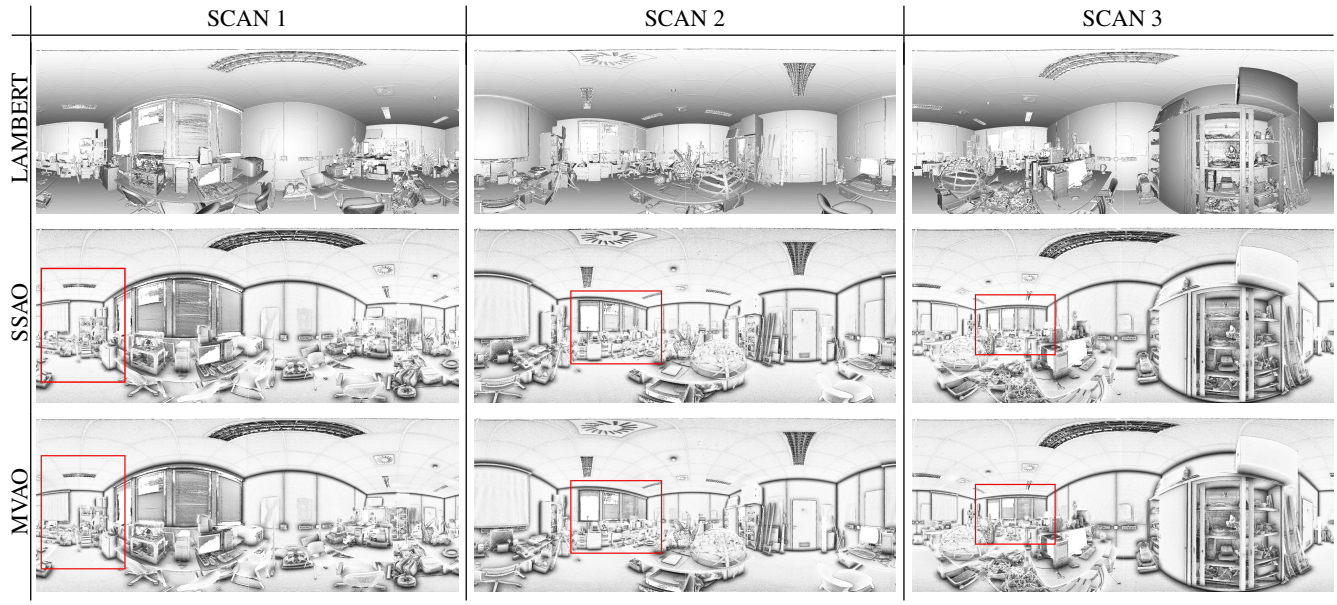

Figure 8: *Results on the scans of the lab dataset: (Top) Lambertian shading; (Center) per depth map SSAO computed using the algorithm in Section 4; (Bottom) MVAO computed using the algorithm in Section 5. The red rectangles highlight an area with differences between the SSAO and the MVAO.*

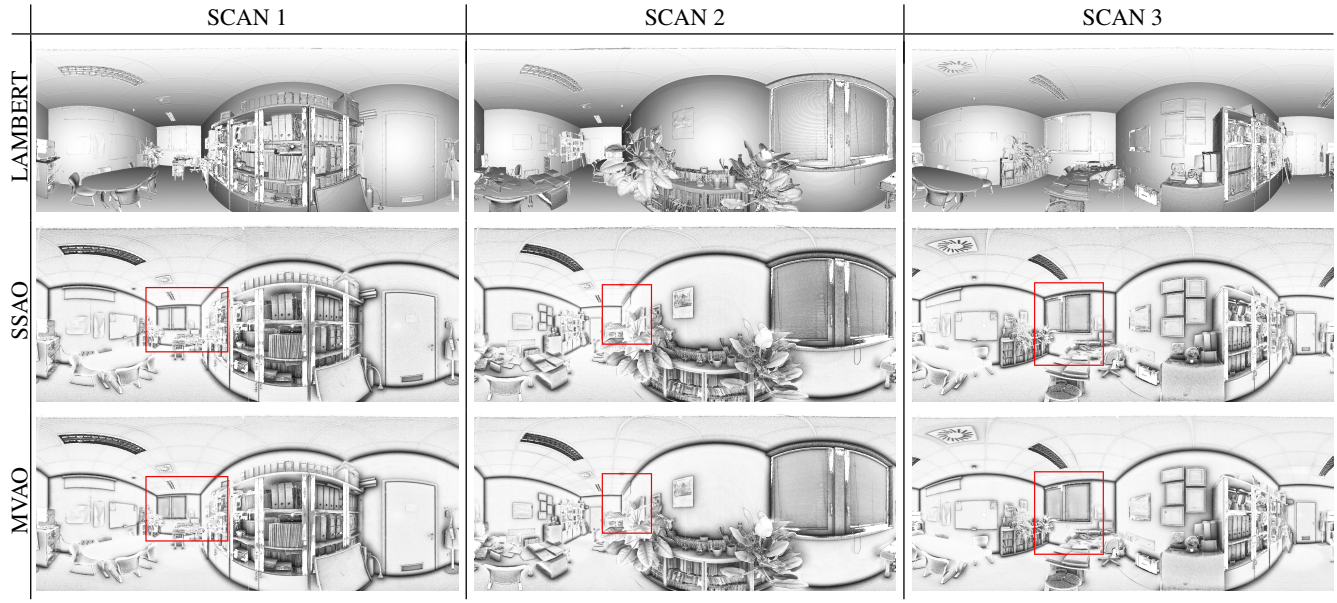

Figure 9: *Results on the scans of the office dataset: (Top) Lambertian shading; (Center) per depth map SSAO computed using the algorithm in Section 4; (Bottom) MVAO computed using the algorithm in Section 5. The red rectangles highlight an area with differences between the SSAO and the MVAO.*

4 and 6 show the MVAO results obtained by varying the normalization threshold d_{max} and the number N of samples for scans. The minimum normalization threshold d_{min} is fixed and equals to the smallest depth that the scanner can acquired (for the ToF scanner $d_{min} = 1$m). Figure 5 shows the importance of the border weight w_b (Equation 12)in the computation of a smoother AO values near to coverage discontinuities. Table 7 contains the time performance for the computation of the SSAO and MVAO.

The real-time multi view integration algorithm has been tested with the Microsoft Kinect 2. For all the experiments we use a

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

	LAB			OFFICE		
	SCAN 1	SCAN 2	SCAN 3	SCAN 1	SCAN 2	SCAN 3
# vertex	5931968	6044383	6002070	5956802	6047852	6064115
SSAO	27ms	27ms	27ms	26ms	25ms	27ms
MVAO	26ms	28ms	25ms	28ms	26ms	33ms
MVAO 3x3	173ms	201ms	173ms	188ms	184ms	239ms
MVAO 5x5	470ms	541ms	477ms	519ms	476ms	645ms
MVAO 7x7	902ms	1006ms	895ms	1041ms	908ms	1219ms
MVAO 9x9	1402ms	1564ms	1419ms	1585ms	1412ms	1823ms

Table 1: *Test case and performance data. For each scan the table contains the time for the computation of the SSAO and the times for the computation of MVAO with several numbers of samples.*

MVAO grid of 512^3 voxel stored as a sparse grid using the Morton code of each cell. For the SSAO computation we use $r_s = 0.1$m and $t_{max} = 0.2$m, while the minimum d_{min} and maximum d_{max} depth thresholds are set to 0.4m and 3m. The additional material contains a video that shows an acquired sequence with SSAO and MVAO. Figure 10 compares the results for some depth maps of the sequence. Also in this case the algorithm is able to recover for each frame all the main occlusions making the AO more coherent. Then the use of the MVAO grid allows the creation of a softer AO value without the use of a Gaussian blur in the SSAO computation step. The algorithm runs in real time (19-21 FPS).

7.1. CH Complex Dataset

We tested the method on a big dataset of high resolution Time-of-Flight laser scans of a portion of a complex architecture, the Maschio Angioino in Naple. In the specific the dataset shows the internal courtyard of the castle, the "Sala dei Baroni", the gallery at the base of the vault of the "Sala dei Baroni" with 16 windows towards the room, the "Cappella Palatina". Given the huge amount of the input data (about 317M vertices) we implemented an out-of-core version of the algorithm in Section 5 where we save the SSAO map of each scan on the disk and following we reload these maps for the computation of the MVAO of each scan. The computation took 931 seconds (about 2 seconds for the computation on GPU of the SSAO for each scan, 241 seconds for the computation on GPU of the MVAO for each scan and the remaining time for the out-of-core loading of the intermediate data). Figure 11 shows a comparison between the SSAO and the MVAO for some input range scans while Figure 12 shows some renderings of the global point cloud with the MVAO mapped as per-vertex color compared with the same rendering with the SSAO mapped as per-vertex color. A grey-scale different image in the bottom highlights the differences between the two renderings. The video in the additional materials shows a navigation of the castle using only the MVAO mapped as per-vertex color.

8. Conclusions

The proposed Multi-view Ambient Occlusion algorithm is an effective solution to estimate the AO value on 3D scanned data using range map information to robustly integrate the SSAO computed for each single scan. Beyond the choice of operating in range space, the core of the algorithm is a new weighting scheme of the SSAO of each scan that is able to solve the view-dependent artifacts of the SSAO techniques, due to missing occluders non visible in the single view, and to manage in a consistent way the different sampling density among the scans. Another advantage of the method is the capability to work directly with raw scan data (for example a raw point cloud), without the need for a triangular surface. It uses all the available scanning information to estimate the SSAO on the depth map of each scan, where the proximity relation among points is well defined.

The obtained results allow the perception of more details attenuating the AO value in the more noisy regions and adding the contributions of not visible occluders. Two different versions of the algorithm were proposed: a version that works with few high resolution scans by integrating the AO value in each single scan; a real-time variant that works with a continuous stream of depth maps generated by a low resolution depth camera by integrating the AO value in a volumetric grid. The algorithm is quite fast showing real-time performance.

A future extension of the algorithm is the check on the general view coverage of the acquisition, in order to decrease or delete the contribution of the same view acquired in different time. This problem is greater for the real-time version of the algorithm where it is common to stay still in certain positions during the acquisition.

9. Acknowledgment

We thank Marco Callieri for the Maschio Angioino dataset. The research leading to these results was funded by EU FP7 project ICT FET Harvest4D (http://www.harvest4d.org/, G.A. no. 323567).

References

[BM92] BESL P., MCKAY N. D.: A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 14*, 2 (Feb 1992), 239–256. 4

[BS09] BAVOIL L., SAINZ M.: Multi-layer dual-resolution screen-space ambient occlusion. In *SIGGRAPH Talks* (2009), ACM. 2

[BSD08] BAVOIL L., SAINZ M., DIMITROV R.: Image-space horizon-based ambient occlusion. In *ACM SIGGRAPH 2008 talks* (2008), ACM, p. 22. 2

[CCR08] CIGNONI P., CORSINI M., RANZUGLIA G.: Meshlab: an open-source 3D mesh processing system. *ERCIM News 2008*, 73 (2008). 6

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

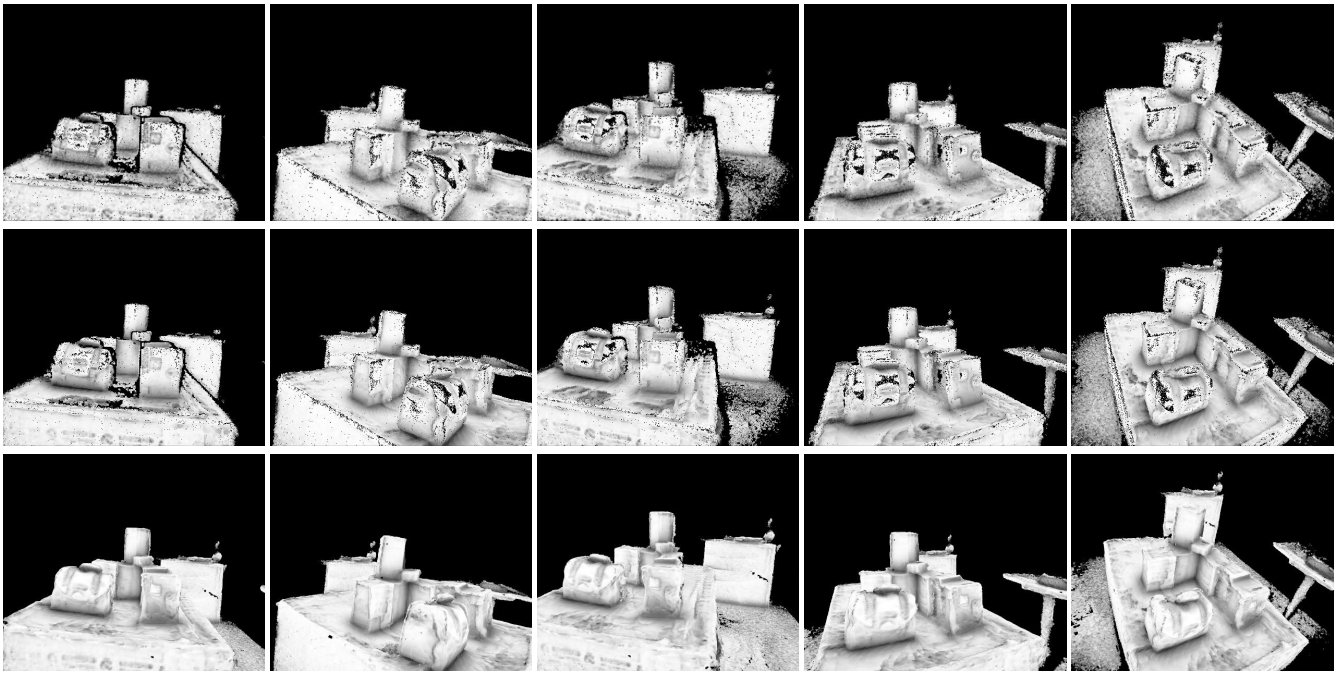

Figure 10: *Comparison of the results for some selected depth maps of the input sequence acquired with the Kinect: (Top) per depth map SSAO; (Center) MVAO computed with the volumetric grid using the algorithm in Section 6; (Bottom) rendering of the final 3D triangular mesh with the MVAO extracted from the volumetric grid.*

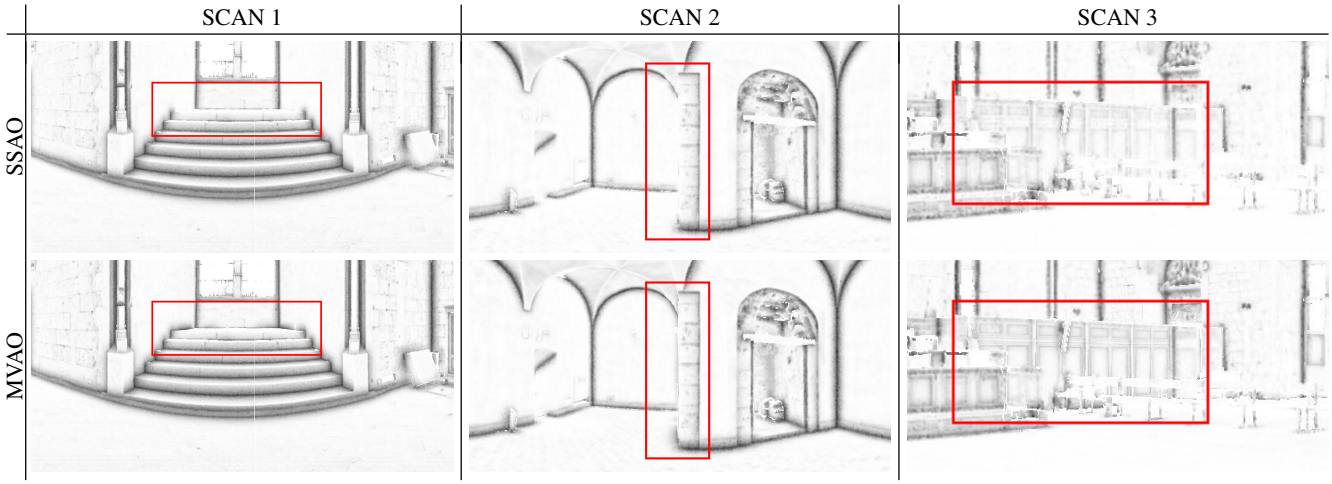

Figure 11: *Results on the scans of the Maschio Angioino in Naple: (Top) per depth map SSAO computed using the algorithm in Section 4; (Bottom) MVAO computed using an out-of-core version of the algorithm in Section 5. The red rectangles highlight an area with differences between the SSAO and the MVAO.*

[HDD*92] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Surface reconstruction from unorganized points. *SIGGRAPH Comput. Graph. 26*, 2 (July 1992), 71–78. 4

[HSEE15] HENDRICKX Q., SCANDOLO L., EISEMANN M., EISEMANN E.: Adaptively layered statistical volumetric obscurance. In *Proceedings of the 7th Conference on High-Performance Graphics* (2015), Doggett M. C., Molnar S. E., Fatahalian K., Munkberg J., Eisemann E., Clarberg P., Spencer S. N., (Eds.), ACM, pp. 77–84. 2

[Lan02] LANDIS H.: Production-ready global illumination. *Siggraph course notes 16*, 2002 (2002), 11. 1, 2

[LB99] LANGER M. S., BÜLTHOFF H. H.: Perception of shape from shading on a cloudy day. *Max Planck Inst Tech Rep 73* (1999), 1–12. 1

[LS10] LOOS B. J., SLOAN P.-P. J.: Volumetric obscurance. In *Proceedings of the 2010 Symposium on Interactive 3D Graphics* (2010), Aliaga D. G., Oliveira M. M., Varshney A., Wyman C., (Eds.), ACM, pp. 151–

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.

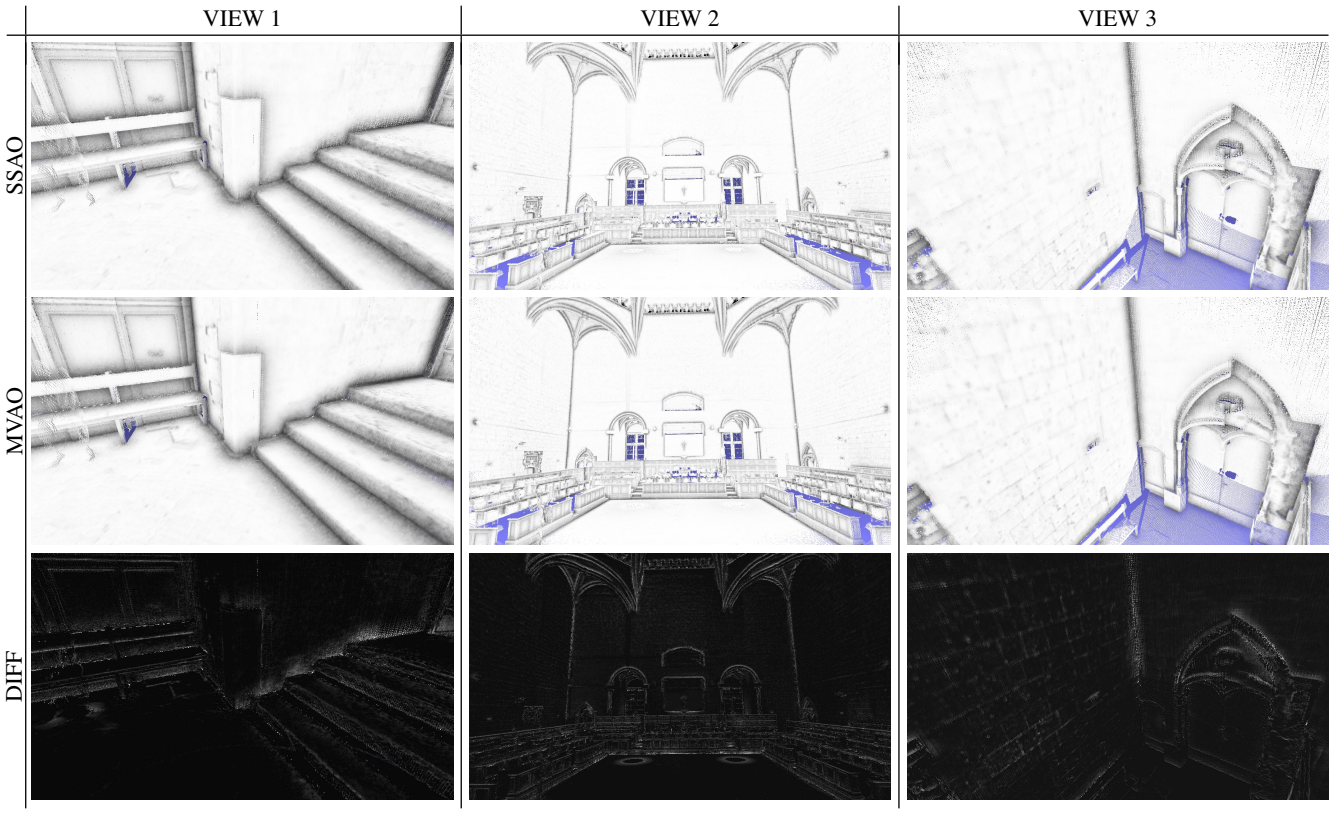

Figure 12: *Rendering of the scans of the Maschio Angioino in Naple: (Top) SSAO mapped as per-vertex color; (Middle) MVAO mapped as per-vertex color; (Bottom) difference image between the two renderings in grey-scale (black means no-difference).*

156. 2

[McG10] McGuire M.: Ambient occlusion volumes. In *Proceedings of the 2010 Symposium on Interactive 3D Graphics* (2010), Aliaga D. G., Oliveira M. M., Varshney A., Wyman C., (Eds.), ACM, pp. 47–56. 2

[MFS09] Méndez-Feliu À., Sbert M.: From obscurances to ambient occlusion: A survey. *The Visual Computer 25*, 2 (2009), 181–196. 2

[Mit07] Mittring M.: Finding next gen: Cryengine 2. In *SIGGRAPH Courses* (2007), McMains S., Sloan P.-P., (Eds.), ACM, pp. 97–121. 2

[MML12] McGuire M., Mara M., Luebke D. P.: Scalable ambient obscurance. In *Proceedings of the Conference on High Performance Graphics 2012* (2012), Dachsbacher C., Munkberg J., Pantaleoni J., (Eds.), Eurographics Association, pp. 97–103. 2

[MOBH11] McGuire M., Osman B., Bukowski M., Hennessy P.: The alchemy screen-space ambient obscurance algorithm. In *Proceedings of the Conference on High Performance Graphics 2011* (2011), Dachsbacher C., Mark W., Pantaleoni J., (Eds.), ACM, pp. 25–32. 2

[NIH*11] Newcombe R. A., Izadi S., Hilliges O., Molyneaux D., Kim D., Davison A. J., Kohi P., Shotton J., Hodges S., Fitzgibbon A.: Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on* (Oct 2011), pp. 127–136. 6

[PCGS15] Pingi P., Corsini M., Ganovelli F., Scopigno R.: Fast and simple automatic alignment of large sets of range maps. *Computers & Graphics 47* (2015), 78–88. 4

[RGS09] Ritschel T., Grosch T., Seidel H.-P.: Approximating dynamic global illumination in image space. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics* (2009), Haines E., McGuire M., Aliaga D. G., Oliveira M. M., Spencer S. N., (Eds.), ACM, pp. 75–82. 2

[RT06] Rong G., Tan T. S.: Jump flooding in GPU with applications to voronoi diagram and distance transform. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics* (2006), Olano M., Séquin C. H., (Eds.), ACM, pp. 109–116. 5

[SA07] Shanmugam P., Arikan O.: Hardware accelerated ambient occlusion techniques on GPUs. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics* (2007), Gooch B., Sloan P.-P. J., (Eds.), ACM, pp. 73–80. 2

[SKUB*10] Szirmay-Kalos L., Umenhoffer T., BalazsToth, Szeci L., Sbert M.: Volumetric ambient occlusion for real-time rendering and games. *IEEE Computer Graphics and Applications 30*, 1 (Sept. 2010), 70–79. 2

[TCM06] Tarini M., Cignoni P., Montani C.: Ambient occlusion and edge cueing for enhancing real time molecular visualization. *Visualization and Computer Graphics, IEEE Transactions on 12*, 5 (2006), 1237–1244. 1

[Tim13] Timonen V.: Line-sweep ambient obscurance. *Computer Graphics Forum 32*, 4 (2013), 97–105. 2

[VPG13] Vardis K., Papaioannou G., Gaitatzes A.: Multi-view ambient occlusion with importance sampling. In *Symposium on Interactive 3D Graphics and Games* (2013), Gopi M., Yoon S.-E., Spencer S. N., Olano M., Otaduy M. A., (Eds.), ACM, pp. 111–118. 2, 4, 5

[ZIK98] Zhukov S., Iones A., Kronin G.: An ambient light illumination model. In *Rendering Techniques 98*. Springer, 1998, pp. 45–55. 1, 2

© 2016 The Author(s)
Eurographics Proceedings © 2016 The Eurographics Association.