

Stochastic visibility in point-sampled scenes

Miles Hansard

<http://www.eecs.qmul.ac.uk/~miles/h/>

School of EECS

Queen Mary, University of London
London E1 4NS

Abstract

This paper introduces a new visibility model for 3-D point-clouds, such as those obtained from multiple time-of-flight or lidar scans. The scene is represented by a set of random particles, statistically distributed around the available surface-samples. Visibility is defined as the appropriate conjunction of occupancy and vacancy probabilities, along any visual ray. These probabilities are subsequently derived, in relation to the statistical scene structure. The resulting model can be used to assign probabilistic visibilities to any collection of scene-points, with respect to any camera position. Moreover, these values can be compared between different rays, and treated as functions of the camera and scene parameters. No surface mesh or volumetric discretization is required. The model is tested by decimating 3-D point-clouds, and estimating the visibility of randomly selected targets. These estimates are compared to reference values, computed by standard methods, from the original full-resolution point-clouds. Applications of the new visibility model to multi-view stereo are discussed.

1 Introduction

Visibility is a fundamental problem in multi-view scene understanding and reconstruction. Simple hidden-surface removal, in the projection of a known 3-D model, can be solved by Z-buffering and related procedures from computer graphics [0, 1, 2, 3, 4]. For more complicated tasks, such as multi-view reconstruction of an *unknown* scene, the representation of visibility remains problematic [5]. There are essentially three approaches, as follows. Firstly, the estimated scene can be represented by a surface mesh, and geometric methods can be used [6, 7, 8]. However, there are many scene-types that cannot be practically reconstructed as meshes (e.g. dense foliage). Secondly, the scene can be represented by a dense voxel grid [9, 10, 11, 12], and visibility can be resolved by labelling the free-space between cameras and surfaces. This approach does not scale easily to large or dynamic scenes (e.g. whole city reconstructions). A third approach is to maintain view-based visibility maps, associated with each camera [13, 14, 15]. The difficulty here is to ensure that all multi-view relationships are consistently represented (e.g. not just the pairwise relationships). A final class of ‘direct’ methods [16, 17] compute visibility via a dual convex hull, but this representation has yet to be adopted in computer vision applications.

This paper proposes a new approach, which assigns a continuous visibility score to all points along each ray. There is a general analogy to voxel-based methods, but without requiring an explicit volumetric representation. In particular, the model proposed here is related to that of Gargallo et al. [18], because both approaches define visibility in relation to a conjunction of ‘vacancies’ between the camera centre and the scene-point. However, the derivation

and representation of these models is completely different; in particular, the present model emerges naturally from an inhomogeneous Poisson process, which is used to model the occupancy of points along each ray. Furthermore, Gargallo et al. use a discrete volumetric representation, whereas the present method works directly from a 3-D point cloud. The tests in section 4 are based on depth-camera data, but future work will consider sparse point-clouds from multi-view keypoint matching. It would then be possible to incorporate photometric visibility information [16, 22] obtained by comparing the 2-D images.

The model developed here has some connections to computer graphics research. Blinn [9] described a scene model consisting of ‘blobs’, somewhat analogous to the Gaussian components used in section 3.1. Lokovic & Veach [23] modelled light penetration, in complex materials such as hair, using an attenuation function similar to that used in section 3.2. Neither of these works were, however, intended to model point-cloud visibility.

1.1 Contributions

The present definition of continuous stochastic visibility, in relation to clearly-defined occupancy and vacancy probabilities, is new. In addition, the following specific contributions are made: §3.1 The probabilistic ‘intersection’ of a visual ray with a Gaussian-mixture scene model is derived. §3.2 A data-adaptive generalization of the Poisson visibility model, for inhomogeneous scenes, is developed. §4.2 A new approach to the evaluation of probabilistic visibility, based on ROC curves, is introduced. §4.3 The performance of the new model is demonstrated, on real data. More generally, this paper is one of the first applications of stochastic geometry to a real 3-D computer vision problem.

2 Scene model

The scene \mathcal{S} is modelled, conceptually, by small particles that are distributed in N ellipsoidal patches \mathcal{S}_k . Let the indicator notation $\mathbb{1}_{\mathbf{p}}$ mean that point $\mathbf{p} = (x, y, z)^\top$ is *occupied* by a scene-particle. In probabilistic terms, the scene is a simple mixture model:

$$\mathcal{S} = \left\{ \mathbb{1}_{\mathbf{p}} : \mathbf{p} \sim \frac{1}{N} \sum_k^N \mathcal{S}_k \right\} \quad (1)$$

where each component \mathcal{S}_k of the mixture has a location \mathbf{q}_k and shape \mathbf{Q}_k , to be defined below (2). Note that the data (e.g. samples from a range scanner) correspond to the points \mathbf{q}_k ; the surrounding particles in are inferred from these. The scene model (1) is an abstraction; it is not necessary to actually generate any particles from it, in order to estimate visibility. Nonetheless, it can be useful to do so, for visualization purposes.

2.1 Patch geometry

Each component \mathcal{S}_k in (1) is defined by a position $\mathbf{q}_k = (x_k, y_k, z_k)^\top$, and a 3×3 covariance matrix \mathbf{Q}_k . The covariance matrix has standard deviations ρ and ε in the tangent-plane and normal direction, respectively. Explicitly, the covariance matrix is a sum of outer-products,

$$\mathbf{Q}_k = \rho^2 (\mathbf{l}_k \mathbf{l}_k^\top + \mathbf{m}_k \mathbf{m}_k^\top) + \varepsilon^2 \mathbf{n}_k \mathbf{n}_k^\top \quad \text{where} \quad \mathcal{Q}_k = \{ \mathbf{q}_k, \mathbf{Q}_k \} \quad (2)$$

contains the location and shape parameters.¹ Each surface normal \mathbf{n}_k can be represented in spherical coordinates, $\mathbf{n}_k = (\sin \theta_k \cos \phi_k, \sin \theta_k \sin \phi_k, \cos \theta_k)^\top$, while \mathbf{m}_k can be any perpendicular unit vector. Then $\mathbf{l}_k \simeq \mathbf{m}_k \times \mathbf{n}_k$ to complete the axes of \mathcal{Q}_k . The ‘radius’ ρ and ‘thickness’ ε parameters can be set globally, so the total dimensionality of the scene model, including the position and orientation of the N patches, is $2 + N(3 + 2)$.

2.2 Viewing geometry

Any view of the scene is modelled as collection of rays, \mathcal{R}_{ij} , each defined by the i -th optical centre \mathbf{c}_i , and the j -th target-point in the scene. Hence a given ray comprises the points

$$\mathbf{p}(t) = \mathbf{c}_i + t\mathbf{u}_{ij} \quad \text{where} \quad \mathcal{R}_{ij} = \{\mathbf{c}_i, \mathbf{u}_{ij}\} \quad (3)$$

and $\mathbf{u}_{ij} \simeq \mathbf{p}_j - \mathbf{c}_i$ is a unit-vector. The parameter $t > 0$ is the distance from \mathbf{c}_i to $\mathbf{p}(t)$. Note that the geometric point $\mathbf{p}(t)$ may or may not be *occupied* by a particle from the scene-model (1) described above. The viewing model (3) is sufficient for the theoretical work in this paper. It can readily be adapted to real cameras, by setting $\mathbf{u}_{ij} \simeq \mathbf{A}_i^{-1}\bar{\mathbf{p}}_j$, where \mathbf{A}_i is the left 3×3 block of the i -th camera matrix, and $\bar{\mathbf{p}}_j \simeq (\bar{x}_j, \bar{y}_j, 1)^\top$ are pixel coordinates in the corresponding image [LS].

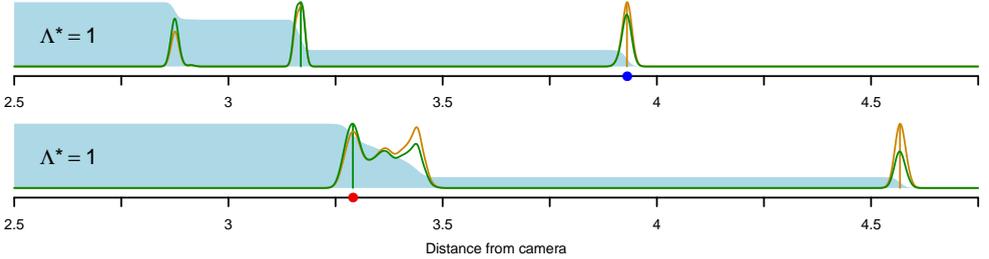


Figure 1: The visibility density $\text{pr}(\dashv t \mid \mathcal{R}, \mathcal{S})$ from (6 & 19) shown in green, for two rays in the evaluation. The target-point of each ray is indicated by a dot, colour-coded by its true state; blue means occluded, and red means visible. Orange curves show the occupancy density $\text{pr}(\mathbb{1}_t \mid \mathcal{R}, \mathcal{S})$ from (5 & 13), while the blue polygon indicates the attenuating vacancy density $\text{pr}(\emptyset_t \mid \mathcal{R}, \mathcal{S})$ from (4 & 16). Vertical lines indicate global maxima along each ray. **Top:** the distant occluded target generates the occupancy maximum (orange), but not the visibility maximum (green). **Bottom:** the nearby visible target generates the visibility maximum (green), but not the occupancy maximum (orange).

3 Visibility model

It is essential to make the right definition of visibility, for point-sampled scenes. In particular, the *event* that point $\mathbf{p}(t) = \mathbf{c} + t\mathbf{u}$ is visible will be denoted $(\dashv t \mid \mathcal{R})$, where the dot is intended to suggest the end of a visual ray. Visibility will now be defined as the conjunction of two events. Firstly, the ray segment $\mathbf{c} + r\mathbf{u}$, where $0 \leq r < t$, should be *vacant*, i.e. free

¹Calligraphic-type is used for sets of parameters, which are identified with the objects that they represent, e.g. $\mathcal{R} = \{\mathbf{p}, \mathbf{u}\}$ is a visual ray, which passes through point \mathbf{p} , in direction \mathbf{u} .

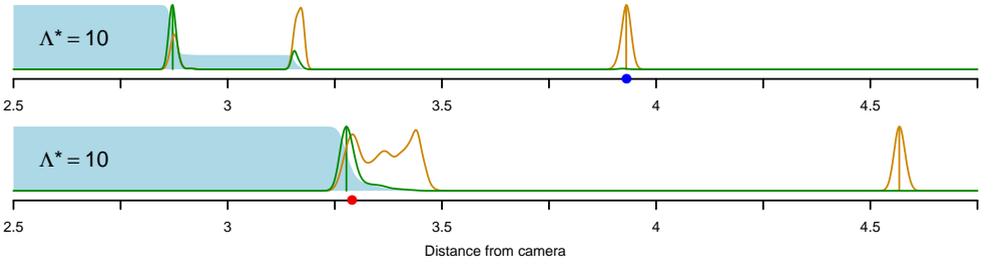


Figure 2: The visibility process (green) tends to emphasize closer points, as the density parameter Λ^* is increased (compare fig. 1). **Top:** the visibility maximum has jumped forward. **Bottom:** the visibility maximum has become biased.

of occluders. Secondly, the point $\mathbf{c} + t\mathbf{u}$ should be *occupied*. The following notation will be used for these events:

$$(\emptyset_t | \mathcal{R}) \Leftrightarrow \text{The ray-segment from } \mathbf{c} \text{ to } \mathbf{c} + t\mathbf{u} \text{ is vacant} \quad (4)$$

$$(\mathbb{1}_t | \mathcal{R}) \Leftrightarrow \text{The point } \mathbf{p} = \mathbf{c} + t\mathbf{u} \text{ is occupied} \quad (5)$$

Hence the *probability* of the point $\mathbf{p}(t) = \mathbf{c} + t\mathbf{u}$ on ray \mathcal{R} being visible in scene \mathcal{S} is the *product* of the vacancy and occupancy probabilities:

$$\text{pr}(\neg \circ t | \mathcal{R}, \mathcal{S}) = \text{pr}(\emptyset_t | \mathcal{R}, \mathcal{S}) \times \text{pr}(\mathbb{1}_t | \mathcal{R}, \mathcal{S}) / |\mathcal{R} \cap \mathcal{S}|. \quad (6)$$

The positive scalar $|\mathcal{R} \cap \mathcal{S}|$ is a normalizing constant, which ensures that something is always seen along the ray; $\int_0^\infty \text{pr}(\neg \circ t | \mathcal{R}, \mathcal{S}) dt \equiv 1$. It is natural, as will become clear, to view $\text{pr}(\emptyset_t | \mathcal{R}, \mathcal{S})$ as an ‘attenuation’ of $\text{pr}(\mathbb{1}_t | \mathcal{R}, \mathcal{S})$ in (6). Hence the two terms will be treated in reverse order, in the following subsections. Real examples of the occupancy, vacancy, and visibility densities are shown in figures 1 and 2.

3.1 Occupancy process

The occupancy $\mathbb{1}_t$ from (5) will now be developed, as a stochastic process. The probability that a point \mathbf{p} occurs in the neighbourhood of a known point \mathbf{q} is determined by the covariance matrix \mathbf{Q} , and therefore has an elliptically contoured distribution. It will be useful, for later derivations, to express this in the general form

$$\text{pr}(\mathbf{p} | \mathcal{Q}) = |\mathbf{Q}|^{-\frac{1}{2}} G_3(|\mathbf{p}|_{\mathcal{Q}}^2) \quad (7)$$

where $G_d(x)$ is the density function in d dimensional space, $|\mathbf{Q}|$ is proportional to the volume of the patch, and $|\mathbf{p}|_{\mathcal{Q}}$ is the Mahalanobis distance from point \mathbf{p} to patch \mathcal{Q} . The distance is defined, in relation to the patch centre \mathbf{q} and covariance \mathbf{Q} , as

$$|\mathbf{p}|_{\mathcal{Q}}^2 = (\mathbf{p} - \mathbf{q})^\top \mathbf{Q}^{-1} (\mathbf{p} - \mathbf{q}). \quad (8)$$

The density functions will be Gaussian, in order to ensure that the patches are well-localized by the exponential decrease:

$$G_d(x) = (2\pi)^{-\frac{d}{2}} \exp(-\frac{1}{2}x). \quad (9)$$

This definition is made to be compatible with the probability and distance functions in (7) and (8), respectively. Note that the mixture components in (1) can be thought of as the pairing of a density function with a mean and covariance: $S_k = \{G_3(\cdot), Q_k\}$.

The intersection of a ray with a surface-patch is generalized here, to mean the *closest approach* of point on the ray to the patch, with respect to the Mahalanobis distance (8). This implies tangency of the ray \mathcal{R} to an iso-contour of the quadric \mathbf{Q}^{-1} . The direction \mathbf{u} of the ray must be perpendicular to the normal vector $\mathbf{v} = \mathbf{Q}^{-1}(\mathbf{c} + t\mathbf{u} - \mathbf{q})$, at the point of tangency, where $t = \mu$. This is obtained by solving $\mathbf{u}^\top \mathbf{v} = 0$, which gives

$$\mu = -\sigma^2 \mathbf{u}^\top \mathbf{Q}^{-1}(\mathbf{c} - \mathbf{q}) \quad \text{where} \quad \sigma^2 = 1/(\mathbf{u}^\top \mathbf{Q}^{-1} \mathbf{u}). \quad (10)$$

These definitions can now be used to develop the Mahalanobis distance $|\mathbf{p}|_{\mathcal{Q}}^2$ in relation to the point μ , with \mathbf{p} constrained to lie on the ray \mathcal{R} , as follows.

$$\begin{aligned} |\mathbf{p}|_{\mathcal{Q}}^2 &= (\mathbf{c} + t\mathbf{u} - \mathbf{q})^\top \mathbf{Q}^{-1}(\mathbf{c} + t\mathbf{u} - \mathbf{q}) \\ &= t^2 \mathbf{u}^\top \mathbf{Q}^{-1} \mathbf{u} + 2t \mathbf{u}^\top \mathbf{Q}^{-1}(\mathbf{c} - \mathbf{q}) + |\mathbf{c}|_{\mathcal{Q}}^2 \\ &= (t - \mu)^2 / \sigma^2 + \tau^2 \quad \text{where} \quad \tau^2 = |\mathbf{c}|_{\mathcal{Q}}^2 - \mu^2 / \sigma^2. \end{aligned} \quad (11)$$

The density (7) can therefore be expressed, via (11), as a 1-D Gaussian function of the ray-parameter t , for point $\mathbf{p}(t)$. It follows from (7) that the occupancy probability is

$$\begin{aligned} \text{pr}(\mathbb{1}_t | \mathcal{Q}, \mathcal{R}) &= |\mathbf{Q}|^{-\frac{1}{2}} G_3(|\mathbf{p}(t)|_{\mathcal{Q}}^2) \\ &= w G_1((t - \mu)^2 / \sigma^2) \quad \text{where} \quad w = G_2(\tau^2) / |\mathbf{Q}|^{\frac{1}{2}} \end{aligned} \quad (12)$$

is a constant weight for each patch \mathcal{Q} , and ray \mathcal{R} . Note that the parameters μ , σ and τ are also obtained from the known patch and ray, via (10) and (11). The complete occupancy probability, along ray \mathcal{R} , is a weighted sum of 1-D Gaussians,

$$\begin{aligned} \text{pr}(\mathbb{1}_t | \mathcal{R}, \mathcal{S}) &= \frac{1}{N} \sum_{\mathcal{Q} \in \mathcal{S}} \text{pr}(\mathbb{1}_t | \mathcal{Q}, \mathcal{R}) \\ &= \frac{1}{N} \sum_k^N w_k G_1((t - \mu_k)^2 / \sigma_k^2) \end{aligned} \quad (13)$$

over all patches \mathcal{Q} in the scene \mathcal{S} . This sum will be dominated by patches that are both close and perpendicular to the ray.

3.2 Vacancy process

It has been shown elsewhere [R3, R4, R2] that if the scene consists of *randomly* distributed particles, of radius ε , then it can be modelled as a volumetric Poisson distribution. It follows directly that the probability of point $\mathbf{p}(t)$ being non-occluded is

$$\text{pr}(\emptyset_C) = \exp(-\lambda |C|) \quad (14)$$

where $|C| = \pi \varepsilon^2 t$ is the volume of the cylinder between the \mathbf{p} and the optical centre. The *intensity* parameter λ represents the expected number of points per unit volume. This model

will now be generalized to the case of non-random scenes, as defined by point-sampled surfaces.

Intuitively, if the cylinder \mathcal{C} intersects a sampled surface, then its volume should be locally inflated, in order for (14) to show a strong decrease in the vacancy probability. This constitutes an *inhomogeneous* Poisson model [9]. More specifically, the local volume can be taken proportional to $\text{pr}(\mathbb{1}_t \mid \mathcal{R}, \mathcal{S}) \times dt$. Hence a generalized volume of \mathcal{C} is obtained by integrating the Gaussian mixture (13) along the ray \mathcal{R} . The integrand is a sum of non-negative terms, so the integral can be moved inside the sum (Fubini’s theorem), and then performed, as follows:

$$\begin{aligned} \Lambda(t) &= \int_0^t \text{pr}(\mathbb{1}_r \mid \mathcal{R}, \mathcal{S}) \, dr \\ &= \frac{1}{N} \sum_k^N w_k H_1((t - \mu_k)/\sigma_k) \end{aligned} \quad (15)$$

where $H_1(x)$ is the standard cumulative density of the Gaussian $G_1(x)$. The vacancy probability can now be defined by combining (15) with the Poisson model (14). However, the argument of the exponential should be obtained from an expected count, rather than from a probability. Hence a dimensionless scaling $\eta > 0$ is introduced, and:

$$\begin{aligned} \text{pr}(\emptyset_t \mid \mathcal{R}, \mathcal{S}) &= \exp(-\eta \Lambda(t)) \\ &= \frac{1}{N} \prod_k^N \exp\left(-\eta w_k H_1((t - \mu_k)/\sigma_k)\right). \end{aligned} \quad (16)$$

Observe that the product is over ‘soft’ step-functions $H_1(\cdot)$, which test for occluders in front of t . It is convenient to define η in relation to a more intuitive parameter: the average density Λ^* of scene-points along any ray. If the length of a particular ray is T_j , then

$$\eta_j = \Lambda^* / \Lambda(T_j). \quad (17)$$

The above procedure effectively sets the expectation of the negative log vacancy (i.e. the effective total occupancy), over all rays:

$$\langle -\log \text{pr}(\emptyset_T \mid \mathcal{R}, \mathcal{S}) \rangle_{\mathcal{R} \mid \mathcal{S}} \equiv \Lambda^*. \quad (18)$$

Although, in principle, $T_j = \infty$ for (17), the density must fall to zero very quickly after a certain point, assuming that the captured point-cloud is finite. In practice, a good choice for the limit is a point in the tail of the most distant 1-D Gaussian on the ray, $T_j = \max_j(\mu_j + 3\sigma_j)$. There will be no visible points beyond this limit, in practice.

3.3 Visibility process

The probability (6) of seeing point $\mathbf{p}(t) = \mathbf{c} + t\mathbf{u}$, given the ray \mathcal{R} and scene \mathcal{S} , can finally be expressed as the product of the vacancy and occupancy probabilities:

$$\begin{aligned} \text{pr}(-\circ t \mid \mathcal{R}, \mathcal{S}) &\propto \text{pr}(\emptyset_t \mid \mathcal{R}, \mathcal{S}) \times \text{pr}(\mathbb{1}_t \mid \mathcal{R}, \mathcal{S}) \\ &= \frac{\exp(-\eta \Lambda(t))}{|\mathcal{R} \cap \mathcal{S}|} \sum_k^N w_k G_1((t - \mu_k)^2 / \sigma_k^2). \end{aligned} \quad (19)$$

The parameter η effectively sets the sensitivity to occluding points, via definition (17); for example, the simple occupancy model (13) is obtained as $\eta \rightarrow 0$.

Note that the last expression in (19) is normalized by the scalar $|\mathcal{R} \cap \mathcal{S}|$, which represents the total ‘intersection’ of the ray with the scene density, and ensures that something is always seen; $\int_0^T \text{pr}(\neg o_t | \mathcal{R}, \mathcal{S}) dt \equiv 1$. The constant is obtained by integrating along the whole ray:

$$|\mathcal{R} \cap \mathcal{S}| = \int_0^T \text{pr}(\emptyset_t | \mathcal{R}, \mathcal{S}) \text{pr}(\mathbb{1}_t | \mathcal{R}, \mathcal{S}) dt$$

where T is obtained from the tail of the furthest Gaussian, as in sec. 3.2. In practice $|\mathcal{R} \cap \mathcal{S}|$ can easily be obtained by numerical integration, given that the integrand is a smooth function. A standard Gauss-Kronrod routine [25] was used in all experiments reported here.

Note that $\text{pr}(\neg o_t | \mathcal{R}, \mathcal{S})$ in (19) is a probability *density*, and therefore does *not* have a pre-determined range. This means that a hard visible/occluded classification, if required, cannot be based on an *a priori* threshold. Nonetheless, a threshold can be determined from other criteria, as described in 4.2, below.

4 Experiments

It is hypothesized that the model (19) can be used to estimate the visibility of a given target, based on a point-cloud representation of the scene. In particular, the point-cloud may be relatively sparse, and may not include the target itself. The hypothesis is tested by first determining the *reference visibilities* of a collection of targets in a *high resolution* point-cloud, using standard methods [25]. The point-cloud is then decimated, and the *estimated visibilities* of the targets, according to the present model, are compared to their reference values.

4.1 Procedure

There are two essential requirements on the data, as follows. Firstly, the scene must have been captured from many different viewpoints, in order to be sufficiently challenging; i.e. a typical ray should intersect the scene many times. Secondly, the visible surfaces must be densely sampled, in order to simplify the computation of reference visibilities. The present evaluation is based on the Washington RGB-D Scenes Dataset (V2) [27]. This set contains dense scans of indoor scenes, based on 3-D registration of RGB-D video. Two different scenes, A and B (09.ply and 14.ply, shown in fig. 3) were selected. Twelve optical centres \mathbf{c}_i were positioned in a ring, at head-height, around the scene (at 30° separations, as in fig. 4). Hence one evaluation of the target-set yields $12 \times 100 = 1200$ visibility estimates.

Standard PCL procedures [27] were run on the input data, to remove outlying points, and to estimate the normal vectors \mathbf{n}_k for (2). The data were then voxelized to one point per δ^3 , where $\delta = 1\text{cm}$. This step ensures that randomly-selected targets are not biased to be in overlapping scan-regions. A set of 100 target-points was then selected at random, from each point-cloud. Reference visibilities were computed by a discrete ray-tracing procedure [9, 25]. Specifically, a point $\mathbf{p}_{\text{target}}$ was labelled *occluded* if another point was present in the cylinder of radius $\delta/2$, connecting $\mathbf{p}_{\text{target}}$ to a camera centre. Occluders within 2δ of the target were ignored, because these are invariably due to noise on the target surface.

The clouds were finally re-voxelized, for testing, to one point per δ_{sub}^3 , where δ_{sub} was chosen to yield only 10% of the original points (not including the targets). This procedure

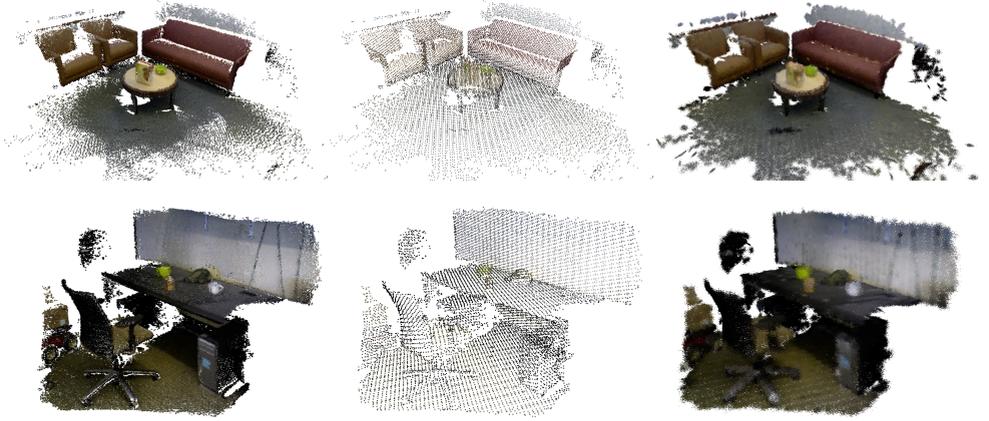


Figure 3: **Top-left:** Reference point-cloud for scene A, after outlier-removal and voxel-based re-sampling (to even-out the data). **Top-centre:** Decimated point-cloud A, containing 16483 points (10% of the original data), used for testing. **Top-right:** The scene model obtained by ‘up-sampling’ the decimated cloud, according to the mixture model described in the paper. Each patch was sampled 100 times, and sample-colours were inherited from the given centre-points (NB this up-sampled rendering is purely for illustration). **Bottom left, centre, right:** the original, decimated, and up-sampled clouds for scene B. The decimated cloud contains 15189 points, in this case. The original point-clouds are from the Washington RGB-D Scenes Dataset (V2) [24].

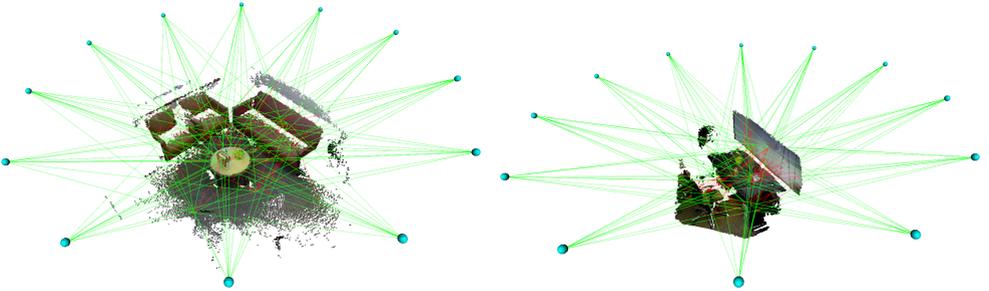


Figure 4: Camera configuration for the experiments in scene A (**left**), and scene B (**right**). Twelve optical centres were positioned in a ring, at head-height, around each scene (at 30° separations). The rays for ten targets are shown (100 targets were used for each evaluation). Green segments are vacant; red segments contain one or more intersections.

left 16483 points in scene A, and 15189 points in scene B. The patch radius and thickness parameters (2) were set to $\rho = \delta_{\text{sub}}/2$ and $\varepsilon = \rho/4$, respectively.

4.2 Criteria

Visibility errors may be false positives (accepted occluded points) or false negatives (rejected visible points), and the underlying ratio of visible / occluded points will depend on the nature of the scene. Furthermore, as mentioned in 3.3, there is no *a priori* threshold for the density $\text{pr}(\neg t \mid \mathcal{R}, \mathcal{S})$ in (19). These issues can be addressed by the analysis of ROC visibility

plots, which show the result of all possible thresholds. The area under the curve (AUC) gives a convenient numerical summary of the performance, corresponding to the probability of mis-classifying a random target as visible / occluded. This measure is also invariant to the underlying ratio of instances [14], which makes it possible to compare across different scenes (and viewpoints).

4.3 Results

The results of the ROC analysis, for $12000 = 2 \times 5 \times 1200$ visibility estimates, are shown in fig. 5. It is clear that the pure occupancy model, obtained by setting $\Lambda^* = 0$ in (17), already performs quite well (orange curves). The area under the orange curve in fig. 5 is 0.81 for scene A, and 0.80 for scene B. This result can be understood by noting that occupancy is a necessary condition for visibility. Furthermore, weak occupancy is correlated with occlusion, because the normalization $\int \text{pr}(\neg t \mid \mathcal{R}, \mathcal{S}) dt \equiv 1$ implies strong occupancy *elsewhere* on the ray (but perhaps behind the target).

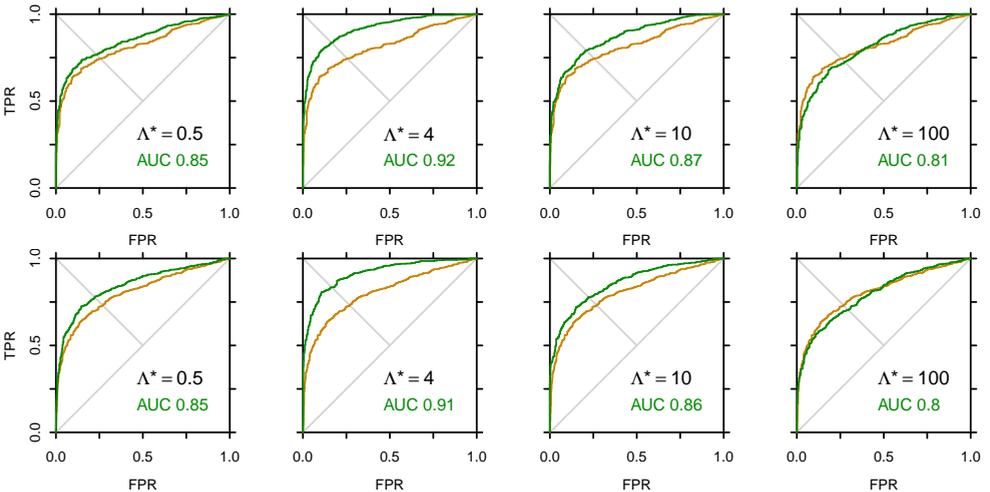


Figure 5: **Top row:** ROC plots of visibility classification in scene A. Green curves show the consequences of making 1200 visible (+) or occluded (−) decisions, by applying a variable threshold to $\text{pr}(\neg t \mid \mathcal{R}, \mathcal{S})$ in (19). All points are classified as occluded at the start of each curve, and visible at the end of each curve. Results for four density values Λ^* are shown in relation to the orange occupancy curve for $\Lambda^* = 0$. Performance is summarized by the area under the curve (AUC) in each case. **Bottom row:** Similar results are obtained for scene B.

It is also apparent from the results in fig. 5 that incorporating the vacancy process (16) leads to a significant improvement in performance. In particular, for $\Lambda^* = 4$, AUC values greater than 0.9 are obtained for both scenes. This result gives strong support to the visibility model presented here.

As $\Lambda^* \rightarrow 0$, it is clear that the visibility model reduces to the occupancy model, as suggested by the $\Lambda^* = 0.1$ curves on the left. More interesting behaviour can be seen at the other extreme, which corresponds to suppressing all but the *closest* point on each ray. Classifying the remaining points as visible is a reasonable strategy, as can be seen in the top-right of the $\Lambda^* = 100$ curves. However, the lower parts of these curves have *worse* performance than the

pure occupancy model (i.e. the green curve is below the orange curve). Indeed, the corresponding strategy does not make sense: take only the closest points, but then require very strong occupancy, in order to classify them as visible.

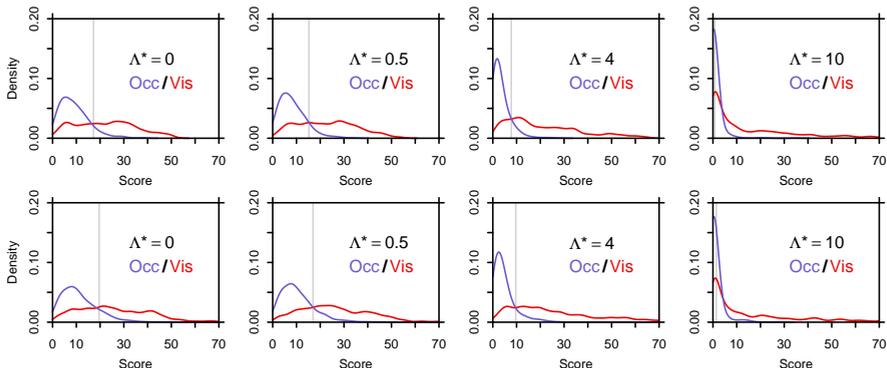


Figure 6: **Top:** Kernel density (bandwidth = 2) estimates of class-conditional visibility probabilities, in scene A. The vertical line represents an optimal threshold, obtained from the ROC analysis. A good separation (AUC 0.92) is obtained with density parameter $\Lambda^* = 4$, as in fig. 5. **Bottom row:** Similar results are obtained for scene B, for which $\Lambda^* = 4$ also gives a good separation (AUC 0.91).

The corresponding visibility score distributions are shown in fig. 6. It is clear from these plots that increasing the density parameter Λ^* has the effect of down-weighting the occluded points (blue curves). It is interesting to see how similar the distributions are, between the two different scenes.

5 Discussion

A new model of visibility and occlusion has been developed, and evaluated on 3-D point-cloud data. Some possible applications, in multi-view stereo, were suggested in the introduction. This direction seems particularly promising, because the visibility densities are already parameterized by the scene and camera variables. One approach would be to use sparse keypoint-matches to define the visibility model, and then to augment the point cloud via photo-consistency maximization with respect to the images [20, 81, 83].

The current model will be tested, in future, on a wider variety of lidar / depth-camera point clouds, including those representing outdoor scenes. It will also be interesting to consider locally-defined patch sizes and shapes, rather than using global radius and thickness parameters in (2). The experiments reported here were based on brute-force algorithms, which would not scale to much bigger scenes. It will therefore be necessary to develop a more efficient implementation, using established data-structures from computer graphics [4]. In particular, it should be possible to use the known adjacency of neighbouring rays, in each lidar or depth camera scan, in order to develop more efficient algorithms.

In summary, the model presented here goes beyond Z-buffering and depth-sorting, towards a proper statistical understanding of visibility and occlusion. These ideas, in conjunction with existing models of colour and surface variation, might also contribute to future theories of natural image-generation.

References

- [1] M. Agrawal and L. S. Davis. A probabilistic framework for surface reconstruction from multiple images. In *Proc. CVPR*, pages 470–476, 2001.
- [2] B. Alsadik, M. Gerke, and G. Vosselman. Visibility analysis of point cloud in close range photogrammetry. In *ISPRS Annals*, volume II-5, pages 9–16, 2014.
- [3] M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, and C. Silva. State of the Art in Surface Reconstruction from Point Clouds. In *Proc. Eurographics 2014*, volume 1, pages 161–185, 2014.
- [4] J.F. Blinn. A generalization of algebraic surface drawing. *ACM Trans. Graphics*, 1(3): 235–256, 1982.
- [5] M. Botsch, A. Wiratanaya, and L. Kobbelt. Efficient high quality rendering of point sampled geometry. In *Proc. Eurograph. Workshop on Rendering*, pages 53–64, 2002.
- [6] D.R. Cox. Some statistical methods connected with series of events. *J. Royal Statistical Society B*, 17(2):129–164, 1955.
- [7] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proc. SIGGRAPH*, pages 303–312, 1996.
- [8] A. Delaunoy and E. Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3D reconstruction problems dealing with visibility. *Int. J. Computer Vision*, 95(2):100–123, 2011.
- [9] F. Duguet and G. Drettakis. Robust epsilon visibility. *ACM Trans. Graphics*, 21(3): 567–575, 2002.
- [10] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.
- [11] P. Gargallo, E. Prados, and P. Sturm. Minimizing the reprojection error in surface reconstruction from images. In *Proc. ICCV*, pages 1–8, 2007.
- [12] P. Gargallo, P. Sturm, and S. Pujades. An occupancy-depth generative model of multi-view images. In *Proc. ACCV*, pages 373–383, 2007.
- [13] M. Hansard. Binocular projection of a random scene. In *Proc. BMVC*, pages 1–11, 2012.
- [14] M. Hapala and V. Havran. Review: Kd-tree traversal algorithms for ray tracing. *Computer Graphics Forum*, 30(1):199–213, 2011.
- [15] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [16] C. Hernandez, G. Vogiatzis, and R. Cipolla. Probabilistic visibility for multi-view stereo. In *Proc. CVPR*, pages 1–8, 2007.
- [17] J. Huang, A. Lee, and D. Mumford. Statistics of range images. In *Proc. CVPR*, pages 324–331, 1999.

- [18] S. B. Kang and R. Szeliski. Extracting view-dependent depth maps from a collection of images. *Int. J. Computer Vision*, 58:139–163, 2004.
- [19] S. Katz, A. Tal, and R. Basri. Direct visibility of point sets. *ACM Trans. Graphics*, 26(3), July 2007.
- [20] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *Int. J. Computer Vision*, 38(3):199–218, 2000.
- [21] K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3D scene labelling. In *Proc. ICRA*, pages 3050–3057, 2014.
- [22] M. S. Langer and F. Mannan. Visibility in three-dimensional cluttered scenes. *JOSA A*, 29(9):1794–807, 2012.
- [23] T. Lokovic and E. Veach. Deep shadow maps. In *Proc. SIGGRAPH*, pages 385–392, 2000.
- [24] R. Mehra, P. Tripathi, A. Sheffer, and N. J. Mitra. Visibility of noisy point cloud data. *Computers & Graphics*, 34(3):219–230, 2010.
- [25] R. Piessens, E. De Doncker-Kapenga, and C.W. Überhuber. *QUADPACK: A subroutine package for automatic integration*. Springer, 1983.
- [26] J.-P. Pons and J.-D. Boissonnat. A Lagrangian approach to dynamic interfaces through kinetic triangulation of the ambient space. *Computer Graphics Forum*, 26(2):227–239, 2007.
- [27] R. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Proc. ICRA*, pages 1–4, 2011.
- [28] G. Schaufler and H.W. Jensen. Ray-tracing point-sampled geometry. In *Proc. Eurographics Workshop on Rendering*, pages 319–328, 2000.
- [29] S.M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*, volume 1, pages 519–528, 2006.
- [30] C. Strecha, R. Fransens, and L.J. Van Gool. Wide-baseline stereo from multiple views: A probabilistic account. In *Proc. CVPR*, pages 552–559, 2004.
- [31] E. Tola, C. Strecha, and P. Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012.
- [32] G. Vogiatzis, C. Hernandez, P. H. S. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. PAMI*, 29(12):2241–2246, 2007.
- [33] A. Yezzi and S. Soatto. Stereoscopic segmentation. *Int. J. Computer Vision*, 53(1):31–43, 2003.
- [34] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Surface splatting. In *Proc. SIGGRAPH*, pages 371–378, 2001.