A Generative Model for Online Depth Fusion

Oliver J. Woodford[†], George Vogiatzis[‡]

[†]Toshiba Research Europe Ltd., Cambridge, UK
[‡]Aston University, Birmingham, UK

Abstract. We present a probabilistic, online, depth map fusion framework, whose generative model for the sensor measurement process accurately incorporates both long-range visibility constraints and a spatially varying, probabilistic outlier model. In addition, we propose an inference algorithm that updates the state variables of this model in linear time each frame. Our detailed evaluation compares our approach against several others, demonstrating and explaining the improvements that this model offers, as well as highlighting a problem with all current methods: systemic bias.

1 Introduction

Today a plethora of depth measuring technologies are available: traditional and active stereo, sonar, time-of-flight cameras and laser scanners, to name but a few. They all measure the distance from the sensor to the closest surface along a ray, or grid of rays, but have varying noise, outlier, speed and resolution characteristics. These sensors are being used in an ever growing number of vision applications, such as vision for vehicles and scene reconstruction, where measurements are made from many different locations (here assumed known), sometimes using multiple sensor types; usually an estimate of scene geometry which fuses all these measurements is required.

The majority of papers on this subject—depth map fusion—fall into one of two categories: online methods which receive depth measurements over time and output the current depth estimate at each time step, and offline methods which fuse a set of depth maps to give a single geometry estimate. This paper falls into the former category. Our contributions are to accurately incorporate both visibility constraints and a variable outlier ratio into the model, while maintaining a complexity linear in the number of state variables.

Visibility constraints. Since the sensors considered see only the first surface along a ray, their measurements provide additional clues as to the location of surfaces: given the surface that generated the measurement, there can be no surface along the ray from that point towards the sensor (the "free space" constraint), but along the ray from that point away from the sensor there may be further surfaces. This implies a dependency between the likelihood of a measurement at one point along a sensor ray and the occupancy of all points along the ray; any model which assumes independence does not, therefore, exploit these constraints. When measurements from different locations are considered, these

dependencies overlap to form a complex Markov Random Field (MRF) with long-range interactions.

The first methods to consider visibility were offline stereo methods, whose photoconsistency costs can be loosely re-interpreted as depth measurement data likelihoods. Some model pixel depths in an MRF, determining visibility from those depths [1, 2], a general approach that has since been very popular, and even made real-time [3]. Others model voxel occupancies, initially considering visibility by visiting unoccluded regions of space first [4], but more recently using complex 3D MRF formulations [5, 6], computing depth as a 3D segmentation between visible and invisible points [5], or computing probabilistic occupancies using long-range ray cliques to model visibility [6]. A drawback to all these offline methods, which treat every input frame equally, is that the MRF structure becomes more complex with larger sets of frames, leading to slower inference. Avoiding this complexity is therefore key for online methods.

Most early online methods, referred to as *inverse sensor models* [7, 8] in the robotics literature, build on the *occupancy grid* of Elfes & Matthies [7], storing a grid/volume of occupancy probabilities, but without any inter-dependency of variables along each ray. A recent evaluation [9] of these approaches highlights Konolige's [8], which incorporates some reasoning on visibility, but based on measurements rather than current occupancies. *Forward sensor models* [10–13] are generative models of the sensor measurement process, which naturally incorporate visibility constraints, but all previous such models have deficiencies. Thrun [13] uses the iterative EM algorithm to update the latent correspondence of measurement to variable, making the method slow. Pathak *et al.* [12] present a closed-form update to probabilities of visibility, and assume that these directly determine occupancy probabilities. As a result, their model has no notion of occluded surfaces; the occupancy updates include an ad-hoc rule to overcome this. Other methods forgo the dependence of prior visibility probability on occupancies along the ray [10, 11], instead using either an ad-hoc distribution based on the occupancy at each point [10] or a delta function on the measurement [11].

Another approach to online depth fusion is to compute a weighted sum of *truncated signed distance functions* (TSDFs) over depth measurements [14–16], usually inferring depth at the zero-isocontour of the resulting volume [14, 15]. This considers visibility implicitly, by downweighting the distance function beyond the surface; indeed, it has been shown to be equivalent to calculating log-odds of visibility assuming a logistic sensor noise model [5].

Robust measurement models. Methods using noisy and cluttered data, such as sonar or stereo measurements, require robust measurement models. Many are able to use any given measurement model, *e.g.* [12, 13], such as those learned from ground truth data; [5] uses a Gaussian plus uniform model. The limitation of these methods is that the exact measurement model must be given prior to inference. Fusing stereo matches, whose clutter characteristics are texture-dependent, hence vary spatially, Vogiatzis *et al.* [17] simultaneously infer depth and an unknown outlier ratio parameter *per pixel*, using a variational scheme, in a framework naturally limited to the multi-view stereo domain.

Geometry priors. Many stereo and depth fusion methods use geometry priors in their inference, *e.g.* [5, 11, 2, 6, 1, 16], encouraging surface smoothness. Here we ignore geometry likelihood, focusing purely on data fusion. We note, however, that a probabilistic output (which we produce) can easily be combined with a geometry prior in a second inference stage.

In the next section we present a new, probabilistic framework for the fusion of depth maps, using a generative model of the sensor measurements (a forward sensor model), within an occupancy volume. Whilst very similar to that of Pathak *et al.* [12], it deals probabilistically with occluded surfaces, and also allows us to associate a different and unknown measurement model *per voxel*, similar to [17]. These unknown measurement models are inferred in parallel with the depth fusion, enabling our method to operate robustly under dramatically different, temporally and even spatially varying, measurement conditions. In §3 we evaluate the improvements resulting from this new model against a range of other methods.

2 A probabilistic approach

Considering, for the time being, a single sensor ray, our model assumes that sensor measurements are generated as a function of the occupancy of points along the ray and the noise properties of the measurement process. In particular, denoting the binary state of occupancy of N discrete points ordered along the ray, starting nearest to the sensor, as $\mathbf{x} = \{x_i\}_{i=1}^{N}$, $x_i \in \{0, 1\}$, we assume, according to the visibility constraint, that a measurement is generated by the first occupied point along the ray (an inlier or true positive), or alternatively by an outlier process which generates false positives. Given the visible point (the true surface), whose index we denote by v, the likelihood of a measurement, y, of the point is given by the measurement noise distribution $\mathcal{M}_v(y)$. If, on the other hand, the measurement is an outlier, its likelihood is given by a clutter distribution, $\mathcal{C}(y)$. Applying these assumptions and distributions, the total likelihood of a measurement, given a state $\mathbf{x}$ and marginalizing over the latent variable v, is

$$p(y|\mathbf{x}, \boldsymbol{\omega}) = \sum_{v=1}^{N+1} p(y|v, \omega_v) p(v|\mathbf{x}), \tag{1}$$

$$p(y|v, \omega) = \omega \cdot \mathcal{C}(y) + (1 - \omega) \cdot \mathcal{M}_v(y), \tag{2}$$

$$p(v|\mathbf{x}) = x_v \prod_{i=1}^{v-1} (1 - x_i), \tag{3}$$

where $\boldsymbol{\omega} = \{\omega_i\}_{i=1}^{N}$, $\omega_i \in [0, 1]$ is a set of outlier ratios, which can also be a single variable or given constant for the sensor ($\omega_i = \omega$)—we investigate all three scenarios here. Note that we have integrated over an additional visible surface index, $v = N{+}1$, which represents the case that no visible surface exists, implying $x_{N+1} = 1$ and $\omega_{N+1} = 1$, since any measurement must be an outlier in this case.

In reality we are given y, as well as a prior distribution, and wish to find a posterior probability over $\mathbf{x}$ and $\boldsymbol{\omega}$. Applying Bayes' rule and choosing an appropriate form of prior, we have

$$p(\mathbf{x}, \boldsymbol{\omega}|y) = \frac{p(y|\mathbf{x}, \boldsymbol{\omega})p(\mathbf{x})p(\boldsymbol{\omega})}{p(y)}, \qquad p(y) = \text{constant}, \tag{4}$$

$$p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i|\gamma_i), \qquad p(x|\gamma) = \gamma^x(1-\gamma)^{1-x}, \tag{5}$$

$$p(\boldsymbol{\omega}) = \prod_{i=1}^{N} p(\omega_i|\alpha_i, \beta_i), \qquad p(\omega|\alpha, \beta) = \frac{\omega^{\alpha-1}(1-\omega)^{\beta-1}}{\mathrm{B}(\alpha, \beta)}, \tag{6}$$

where $\mathrm{B}(\alpha, \beta) = \int_0^1 u^{\alpha-1}(1-u)^{\beta-1}\,\mathrm{d}u$. The prior, consisting of a product of binomial and beta distributions, conveniently exhibits independence over all variables. However, the data likelihood, in particular the visibility constraint of equation (3), makes the variables in $\mathbf{x}$ fully inter-dependent in the posterior.

2.1 A factored approximation

It is impractical to maintain the inter-dependence of variables in $\mathbf{x}$, due to the large state of size 2^N for each ray, and also the MRF complexities which result from intersecting rays. As such, we approximate our posterior with a distribution, $q(\mathbf{x}, \boldsymbol{\omega})$, taking the same form as the prior in equations (5) & (6), but with parameters denoted γ_i', *etc*. This factorization assumes that the state variables are independent, simultaneously reducing the state to size N per ray and avoiding an accumulation of highly connected dependencies. In addition, the state update and reparameterization described in §2.2 are made very simple.

It is common to compute such factored approximations by minimizing the KL-divergence between $p(\mathbf{x}, \boldsymbol{\omega}|y)$ and $q(\mathbf{x}, \boldsymbol{\omega})$, denoted $\mathrm{KL}(q||p)$, or alternatively minimizing $\mathrm{KL}(p||q)$. Here the latter optimization is significantly easier; since $q(\mathbf{x}, \boldsymbol{\omega}) = \prod_{i=1}^{N} q_i(x_i) \prod_{j=1}^{N} q_j(\omega_j)$ is a member of the exponential family, thus:

$$q_i(x_i) = \gamma_i'^{x_i}(1-\gamma_i')^{1-x_i} \qquad = \exp\left(x_i(\ln\gamma_i' - \ln(1-\gamma_i')) + \ln(1-\gamma_i')\right), \tag{7}$$

$$q_j(\omega_j) \propto \omega_j^{\alpha_j'-1}(1-\omega_j)^{\beta_j'-1} \qquad = \exp\left((\alpha_j'-1)\ln\omega_j + (\beta_j'-1)\ln(1-\omega_j)\right), \tag{8}$$

it is a standard result [18, p.505] that minimal $\mathrm{KL}(p||q)$ is achieved matching the expected sufficient statistics, thus:

$$\mathbb{E}_{q(\mathbf{x},\boldsymbol{\omega})}[x_i] = \mathbb{E}_{p(\mathbf{x},\boldsymbol{\omega}|y)}[x_i], \qquad \forall i \in \{1, .., N\}, \tag{9}$$

$$\mathbb{E}_{q(\mathbf{x},\boldsymbol{\omega})}[\ln\omega_i] = \mathbb{E}_{p(\mathbf{x},\boldsymbol{\omega}|y)}[\ln\omega_i], \qquad \forall i \in \{1, .., N\}, \tag{10}$$

$$\mathbb{E}_{q(\mathbf{x},\boldsymbol{\omega})}[\ln(1-\omega_i)] = \mathbb{E}_{p(\mathbf{x},\boldsymbol{\omega}|y)}[\ln(1-\omega_i)], \qquad \forall i \in \{1, .., N\}. \tag{11}$$

Computing $\boldsymbol{\gamma_i'}$ Ordinarily, computing each expectation of equation (9) would require an integration over the 2^{N-1} states of the other variables in $\mathbf{x}$. However,

since the data likelihood term (equation (1)) depends on $\mathbf{x}$ only through v, it can be reduced to the $N+1$ possible values of v by marginalizing out $\mathbf{x}$: $p(v) = \sum_{\mathbf{x}} p(v|\mathbf{x})p(\mathbf{x})$. Given the visibility constraint (equation (3)), which implies $x_i = 0, \forall i < v$, $x_v = 1$ and $x_i \in \{0,1\}, \forall i > v$, and noting that the prior likelihoods of $x_i, \forall i > v$ integrate to one,

$$p(v) = \begin{cases} \gamma_v \prod_{i=1}^{v-1}(1-\gamma_i) & \text{if } v \in \{1,..,N\}, \\ \prod_{i=1}^{N}(1-\gamma_i) & \text{if } v = N+1. \end{cases} \tag{12}$$

The posterior probability of v correctly indexing the visible surface is therefore

$$p(v|y) = \big(\omega_v \cdot \mathcal{C}(y) + (1-\omega_v) \cdot \mathcal{M}_v(y)\big) \cdot \frac{p(v)}{p(y)}, \tag{13}$$

for which we can now compute the constant, $p(y)$, as

$$p(y) = \sum_{v=1}^{N+1} p(y|v,\omega_v) \cdot p(v). \tag{14}$$

When ω_v is a variable (not given), it is computed as the expected value under its prior: $\mathbb{E}_{p(\boldsymbol{\omega})}[\omega_v] = \alpha_v/(\alpha_v+\beta_v)$. Now marginalizing out v, noting that $x_i=1$ if $i=v$ and $x_i=1$ with prior likelihood γ_i if $v<i$, otherwise $x_i=0$, the expectation is

$$\gamma_i' = \mathbb{E}_{p(\mathbf{x},\boldsymbol{\omega}|y)}[x_i] = \underbrace{p(v{=}i|y)}_{v=i} + \underbrace{\gamma_i \textstyle\sum_{j=1}^{i-1} p(v{=}j|y)}_{v<i}. \tag{15}$$

Computing $\boldsymbol{\alpha_i'}$ and $\boldsymbol{\beta_i'}$ Since we know of no closed form solution to the simultaneous equations (10) & (11), and it has been shown [19, p.29] that preserving the two moments, $\mathbb{E}_{q(\mathbf{x},\boldsymbol{\omega})}[\omega_i] = \mathbb{E}_{p(\mathbf{x},\boldsymbol{\omega}|y)}[\omega_i]$ and $\mathbb{E}_{q(\mathbf{x},\boldsymbol{\omega})}[\omega_i^2] = \mathbb{E}_{p(\mathbf{x},\boldsymbol{\omega}|y)}[\omega_i^2]$, is a good approximation with a closed form solution, we choose to solve this system instead. The system is characterized by the two pairs of equations to be equated:

$$\mathbb{E}_{q(\mathbf{x},\boldsymbol{\omega})}[\omega_i] = \frac{\alpha_i'}{\alpha_i'+\beta_i'}, \qquad \mathbb{E}_{q(\mathbf{x},\boldsymbol{\omega})}[\omega_i^2] = \frac{\alpha_i'(\alpha_i'+1)}{(\alpha_i'+\beta_i')(\alpha_i'+\beta_i'+1)} \tag{16}$$

$$p(\omega_i|y) = \sum_{\mathbf{x},\boldsymbol{\omega}_{/i}} p(\mathbf{x},\boldsymbol{\omega}|y) = \underbrace{(S_i + T_i \cdot \omega_i)}_{\text{polynomial } P} \frac{\omega_i^{\alpha_i-1}(1-\omega_i)^{\beta_i-1}}{\mathrm{B}(\alpha_i,\beta_i)}, \tag{17}$$

$$\mathbb{E}_{p(\mathbf{x},\boldsymbol{\omega}|y)}[\omega_i] = \int_0^1 p(\omega_i|y)\omega_i \,\mathrm{d}\omega_i = \left(S_i + T_i \frac{\alpha_i+1}{\alpha_i+\beta_i+1}\right)\frac{\alpha_i}{\alpha_i+\beta_i}, \tag{18}$$

$$\mathbb{E}_{p(\mathbf{x},\boldsymbol{\omega}|y)}[\omega_i^2] = \left(S_i + T_i \frac{\alpha_i+2}{\alpha_i+\beta_i+2}\right)\frac{\alpha_i(\alpha_i+1)}{(\alpha_i+\beta_i)(\alpha_i+\beta_i+1)}. \tag{19}$$

The values of S_i and T_i depend on whether there is an outlier ratio per occupancy variable, or just a single one for the sensor. In the former case

$$S_i = 1 - T_i \cdot \mathbb{E}_{p(\boldsymbol{\omega})}[\omega_i], \qquad T_i = \big(\mathcal{C}(y) - \mathcal{M}_i(y)\big) \cdot \frac{p(v{=}i)}{p(y)}, \tag{20}$$

whilst in the latter case

$$S = 1 - T \cdot \mathbb{E}_{p(\omega)}[\omega], \qquad T = \sum_{i=1}^{N} \big(\mathcal{C}(y) - \mathcal{M}_i(y)\big) \cdot \frac{p(v{=}i)}{p(y)}. \tag{21}$$

However, with just a single outlier ratio, when considering multi-ray sensors the polynomial P in equation (17) becomes much higher order as the ray posteriors are multiplied together. To avoid a costly computation in this case, we compute α' and β' for each ray independently, then average them across all the rays.

2.2 Online fusion

In online approaches, sensor measurements, y^t, are taken at each time step, t, and a current geometry estimate based on all previous measurements is output. We make the standard first-order independence assumptions which imply

$$p(\mathbf{x}^t, \boldsymbol{\omega}^t | y^t, .., y^1) = \frac{p(y^t|\mathbf{x}^t, \boldsymbol{\omega}^t)}{p(y^t)} p(\mathbf{x}^t, \boldsymbol{\omega}^t | \mathbf{x}^{t-1}, \boldsymbol{\omega}^{t-1}) p(\mathbf{x}^{t-1}, \boldsymbol{\omega}^{t-1} | y^{t-1}, .., y^1), \tag{22}$$

which, along with $\mathbf{x}$ being discrete, means ours is a *hidden Markov model*. Most terms in this equation are familiar from the previous section; *e.g.* $p(y^t|\mathbf{x}^t, \boldsymbol{\omega}^t)$ is the data likelihood given by equation (1), and $p(\mathbf{x}^{t-1}, \boldsymbol{\omega}^{t-1} | y^{t-1}, .., y^1)$ is the posterior from the previous time step, whose factored approximation we compute. The new term, $p(\mathbf{x}^t, \boldsymbol{\omega}^t | \mathbf{x}^{t-1}, \boldsymbol{\omega}^{t-1})$, effectively computes the prior for time t from the posterior computed at time $t-1$; this can account for dynamic scenes.

Finally consider an $N \times H \times W$ occupancy grid. This grid can be in the coordinate frame of the current input depth map (s.t. the grid rows and sensor rays are colinear), or alternatively fixed in space, as in [15]. We use the former approach, which, if the sensor moves (or multiple sensors are used), requires a state space reparameterization to each new coordinate frame, but is more efficient overall[1] and ensures that the occupancies are always in view of the current depth map. However, it may lead to pose drift when output depth maps are used in pose estimation, *e.g.* [15].

We assume that we are given a transformation function, $\pi_t : ijk \rightarrow xyz$, which converts coordinates from the frame at time t to that at time $t-1$. Outlier ratio distribution parameters are resampled from the voxel grid of values of the previous time frame, using linear interpolation, as follows:

$$\alpha^t_{ijk} = \alpha'^{t-1}_{\pi_t(ijk)}, \qquad \beta^t_{ijk} = \beta'^{t-1}_{\pi_t(ijk)}. \tag{23}$$

This resampling is the reason we have an outlier ratio per occupancy variable rather than per ray—a per-ray ratio cannot easily be resampled.

[1] The colinear case enables an efficient ray integration, whereby the γ_i, *etc.* of each ray can be updated in two sequential passes along the ray. The fixed-grid case requires a separate ray integration per voxel, making it $\mathcal{O}(N)$ times slower.

Resampling occupancies (*i.e.* γ^t_{ijk} values) is not as trivial, because the computation of $p(v)$ (eqn. (12)) is not sampling-invariant—a naïve resampling which doubles the occupancy resolution will lead to a different shaped distribution. To avoid this we resample the occupancy *density*—each occupancy value is divided by the length of the voxel along the ray, denoted d^t_{ijk}, to compute a density, the new occupancy voxels are sampled, then multiplied by their new length along the ray. This approach can be shown to be sampling-invariant in the limit as voxel lengths tend to zero. In all resamplings, voxels outside the previous grid are initialized to 0.

In addition, to account for motion in scenes, we assume that, between time steps, visible surfaces can disappear with probability ψ, and that new visible surfaces can appear along rays with probability ϕ. To avoid biasing the position of the visible surface, this latter value is divided uniformly in *visibility* (not occupancy) space over all occupancies along a ray, by inverting equation (12), giving

$$p(x^t_i = 1 | x^{t-1}_i = 0) = \frac{\phi}{N - \phi \cdot (i-1)} = \phi_i, \qquad p(x^t_i = 0 | x^{t-1}_i = 1) = \psi. \quad (24)$$

The occupancy update is therefore

$$\gamma^t_{ijk} = \frac{\gamma'^{t-1}_{\pi_t(ijk)}}{d^{t-1}_{\pi_t(ijk)}} \cdot d^t_{ijk} \cdot (1 - \psi - \phi_i) + \phi_i. \quad (25)$$

This update, and those of equation (23), make up the $p(\mathbf{x}^t, \boldsymbol{\omega}^t | \mathbf{x}^{t-1}, \boldsymbol{\omega}^{t-1})$ term of equation (22).

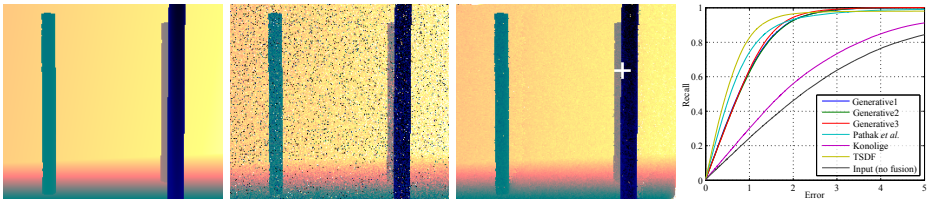
It should be noted that the occupancy prior of equation (24) is viewing direction dependent—a prior that is uniform in visibility space in one direction is not uniform in any other direction. Therefore if the sensor position changes, this can bias the position of the visible surface, regardless of the parameterization of the occupancy grid.

2.3 Computing depth

Above we have described how to compute the occupancy probability of each point in a grid. However, most applications require an actual depth map as output. We compute each depth independently, casting the ray along which the depth is required into the occupancy grid, and outputting the distance to the maximum in the range $[1, N]$ of the posterior visibility distribution, *i.e.* equation (12), but using parameters γ'_i *etc.* The position of each depth output is refined by fitting a quadratic function to the mode.

3 Evaluation

In our evaluation we test our own three methods, namely *Generative1*, which is given ω as an input, *Generative2*, which infers a single ω for the sensor, and



(a) Ground truth (b) Noise added (c) Fused output (d) Error-recall

Fig. 1. Methodology. Each test sequence frame (a) is corrupted with noise (b) per equation (26), then given to the fusion algorithm along with the transformation function, π_t. Given the output of the fusion algorithm (c), error is given by its absolute difference from ground truth. The per pixel disparity errors are used to compute an error-recall graph (d), recall being the proportion of pixels with an error lower than the threshold (x-axis). The graph is plotted up to an error threshold of 5 (our chosen maximum acceptable error), and the area under this graph computed then divided by 5 to give an accuracy score between 0 and 1. This score is computed every frame.

Generative3, which infers an ω_i for each occupancy grid position, and compare them against three methods from the literature, namely those of *Pathak* et al. [12] and *Konolige* [8], and the *TSDF* [14], as implemented in [15].

3.1 Methodology

Our quantitative analysis methodology, outlined in figure 1, is based on synthetic data, so that we have ground truth depth and sensor pose. We parameterize depth by its inverse[2], which we refer to as disparity—the methods' state variables are positioned at integer disparities from 1 to $N = 100$ along each ray. We have generated six sequences of 60–200 frames, similar to figure 1(a), for this evaluation. Each frame of a sequence is corrupted with noise prior to fusion, with the noise applied to each pixel drawn randomly from

$$p(d'|d) = \varpi \cdot \mathcal{U}(d') + (1 - \varpi) \cdot \mathcal{N}(d'; d, \sigma), \tag{26}$$

where d is the real-valued ground truth disparity and d' is the noisy measurement. The uniform distribution, $\mathcal{U}(\cdot)$, represents outliers (ϖ is the outlier ratio), and the normal distribution, $\mathcal{N}(\cdot; \mu, \sigma)$, represents sensor noise. We choose the clutter and measurement distributions of equation (2) to match those distributions, and the given outlier ratio ω to match also, unless otherwise stated. We use $\sigma = 3$ in our tests here, finding that this value didn't change the nature of the results much. This forms the data likelihood term used by all the probabilistic methods, *i.e.* all methods except TSDF, which requires only the disparity estimate and a truncation threshold, for which we use 2σ. The scoring mechanism is described in figure 1.

[2] Noise variances tend to be more uniform over disparity than depth when using stereo-based depth estimates (depending on sensor motion), therefore discretizing regularly over disparity is a more efficient use of state variables in those cases.

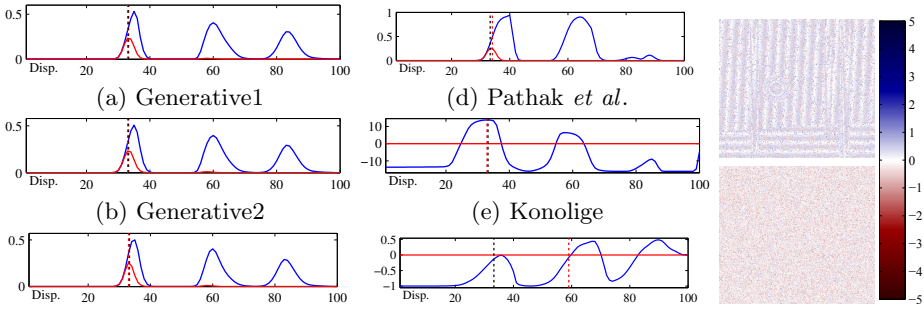


Fig. 2. Occupancies, other states and the effect of discretization. (a)–(f) show, in blue, the occupancies (or equivalent state variable) for the ray marked with a white cross in figure 1(c). The occupancy for (e) Konolige is stored as a log likelihood ratio [8]; the red line denotes occupancy probabilities of 0.5. The red lines in (a)–(d) denote the visibility distribution of equation (12), used to compute depth. Dashed black and red lines mark the ground truth and output disparities respectively. (g), (h) The final error images of the experiment of figure 3(a) for Generative1 and TSDF respectively, showing (g) the artifacts of disparity discretization.

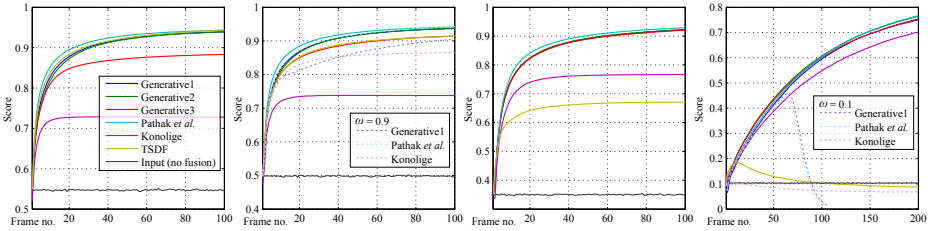
	ϖ	Generative1	Generative2	Generative3	Pathak *et al.*	Konolige	TSDF
Static sensor	0	0.065±0.34	0.069±0.33	0.36±0.62	**0.000**±0.33	1.37±0.25	-0.17±0.23
	0.1	0.064±0.35	0.059±0.34	0.19±0.45	**-0.002**±0.33	1.32±0.25	-0.37±0.31
	0.4	0.090±0.39	0.073±0.39	0.070±0.40	**-0.002**±0.37	1.18±0.33	-7.33±18.0
	0.9	0.60±7.08	**0.24**±6.79	0.45±6.94	0.79±7.91	3.15±11.0	-23.7±20.0
Moving sensor	0	0.35±0.97	0.33±1.09	0.35±1.43	**0.27**±0.86	2.09±8.49	-0.59±3.53
	0.1	0.35±2.13	**0.34**±2.16	0.36±2.20	0.42±2.77	3.69±14.1	-0.69±4.35
	0.4	0.50±3.85	**0.41**±4.09	0.48±3.87	0.74±4.86	6.90±21.1	-1.52±6.87
	0.9	**2.96**±11.1	8.00±18.4	3.39±12.7	3.11±11.1	20.9±32.2	-10.1±12.6

Table 1. Disparity bias. Mean disparity error (± 1 s.d.) for the final frame for two scenes: one with a static sensor, the other with a moving sensor.

Much of the relative performance of methods can be explained by their states, *e.g.* occupancy probabilities, shown for a particular ray (with three seen surfaces) in figure 2. For example, TSDF locates depth as a zero crossing, which can be located more accurately than the maxima used by the other methods, which have discretization artifacts (fig. 2(g),(h)). Also, notice that our three methods (a)–(c) have better defined occupancy maxima than Pathak *et al.*, a result of the latter's incorrect estimation of occupancy from visibility alone. Finally, in the case of (a)–(d), once evidence for a near point has built up, any measurements behind the peak in visibility (red lines) are explained as outliers (if there is an outlier distribution), leading to a systemic *frontal bias*, *i.e.* a bias for closer disparities.

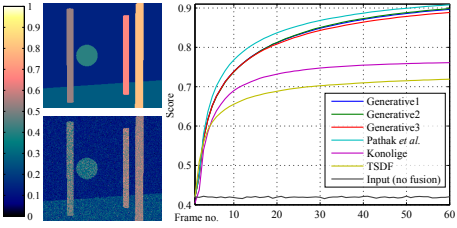
3.2 Results

Static performance. Figure 3 demonstrates how the methods perform on a static scene and sensor. Konolige performs badly with no outliers, and counter-intuitively improves as ϖ, or rather ω (true and given outlier rations respectively; see dotted lines, fig. 3(b)), increases. However, at low outlier ratios it has very



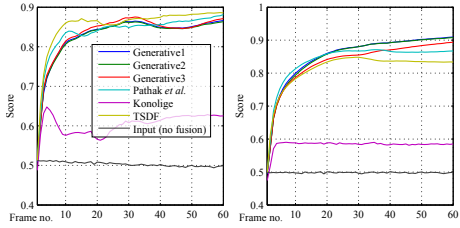
(a) $\varpi = 0$ (b) $\varpi = 0.1$ (c) $\varpi = 0.4$ (d) $\varpi = 0.9$

Fig. 3. Static scene. Scores over time for the six methods, with different values of outlier ratio ϖ. Both scene and sensor are static. In some cases, indicated by dotted lines, the given ratio, ω, differs from the true ratio, ϖ.



(a), (b) (c) Scores

Fig. 4. Variable outlier ratio. (a) (*top*) Per-pixel outlier ratios used on a static scene and static sensor. (b) The outlier ratios of the final output disparities computed by Generative3. (c) The scores over time for the six methods on this sequence.



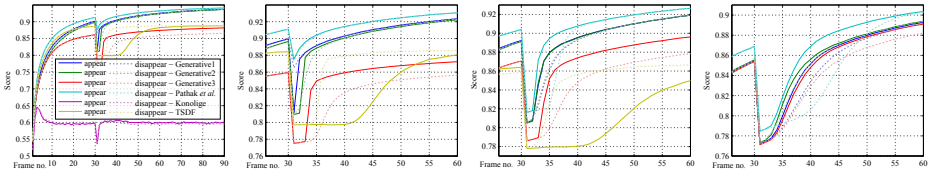
(a) Moving sensor (b) Moving object

Fig. 5. Motion. Scores over time for the six methods, for a static scene and moving sensor (a), and for a static sensor with one object moving in the scene (b). $\varpi = 0.1$ for both sequences.

broad, flat peaks in occupancy (fig. 2(e)), the maxima of which are very sensitive to each new measurement; as ω increases, the clutter distribution competes as an explanation for the measurement, making the peaks narrower and therefore more robust. Generative3 also performs worse than the other methods, a result of frontal bias.

As ϖ increases, TSDF performs increasingly badly; at $\varpi = 0.9$ (fig. 3(d)), whilst the other methods slowly improve their depth estimates over time, TSDF performs worse than just outputting the input disparity. This is a result of the method having no means of accounting for outliers. Pathak *et al*. scores marginally better than the Generative methods; since it uses a stronger assumption, that occupancy derives only from visibility, and not occlusion, when this assumption is true it can be expected to perform better.

If ϖ is overestimated (dotted lines, fig. 3(b)), or varies across rays, as in figure 4, then methods other than TSDF still perform well. Generative3 has no real advantage, even though it is able to estimate ω_i per voxel (fig. 4(b)). However, when ϖ is grossly underestimated (dotted lines, fig. 3(d)), then those methods which cannot estimate ω themselves perform very poorly, again a result of the biasing problem.



(a) $\varpi = 0$ (b) Zoom of (a) (c) $\varpi = 0.1$ (d) $\varpi = 0.4$

Fig. 6. Appearing/disappearing objects. Scores over time for the six methods, with different values of ϖ. Both scene and sensor are static, but two objects either appear or disappear between frames 30 and 31.

Motion performance. With scenes involving motion, ϕ and ψ are estimated from the ground truth data, and applied to all methods except TSDF, for which we set $W_\eta = 10$ (see [15]). Figure 5(a) shows results with a moving sensor and static scene, investigating the effect of surfaces becoming occluded and disoccluded. The methods (Konolige aside, and ignored henceforth) perform very similarly, all showing robustness to those effects.

When objects appear in the scene (figure 6), TSDF does not start to react until W_η frames have passed, and when objects disappear, revealing a background surface, it reacts slightly faster; both effects are independent of ϖ. With $\varpi = 0$ (fig. 6(b)), other methods update to the new disparity much faster, within a frame or two. As ϖ increases (fig. 6(c),(d)), changes can be explained away as outliers, slowing the update, especially the reaction to disappearing surfaces (another impact of frontal bias); Pathak *et al.* suffers particularly here. The Generative approaches' faster reactions mean that they cope better with scene motion, as indicated by figure 5(b).

Systemic bias. Table 1 shows the bias in disparity error for each of the methods on a static and a moving scene. On the static scene the Generative methods have a small frontal bias, caused by nearer occupancies lowering more distant ones due to occlusion modelling. In this case Pathak *et al.* has no significant bias at all, due to its occlusion heuristic—occupancy derives entirely from $p(v)$ up to some significant distance beyond the peak, so only gross outliers can cause a bias. TSDF shows a larger rearward bias, due to the weighting function truncating the TSDF asymmetrically, in favour of more distant measurements. On the moving scene all biases are more significant, with that of Pathak *et al.* being similar in magnitude to those of the Generative methods, a fact that we attribute to the viewpoint-variant occupancy prior.

Monocular stereo. Given a video (and view pose) from a moving camera, we obtain coarse disparity estimates by finding maxima of normalized cross correlation (NCC) scores for 5×5 windows between pairs of frames, similar to [17], and fuse these estimates[3]. We quantitatively evaluate the methods by texture mapping one of our synthetic scenes (with sensor motion) and applying this approach. The results, in figure 7, show TSDF performing significantly better, and

[3] For TSDF we use just the best disparity, while for other methods we compute a data likelihood distribution using all modes of NCC scoring over 0.5.

in particular the Generative methods suffering from a lack of disparity accuracy (fig. 7(b)). This can be seen, from the dominant blue haze in figure 7(c), to be due to frontal bias.

Qualitative results on a real sequence, shown in figure 8, indicate the benefits of our model: fewer disparity artifacts relative to TSDF (see fig. 8(g), center and top left), and fewer artifacts at occlusion boundaries (right of dome) compared to Pathak *et al*. However, figures 8(c,d) suggest the relative biases of the three methods persist on real data too.

KinectFusion. We implemented Generative1 as the fusion strategy within an open source implementation [20] of KinectFusion [15]. Figure 9 compares our result with those of Pathak *et al*. and the original algorithm. We use a $384 \times 960 \times 720$ volume, providing a border around each 640×480 frame; the original TSDF approach uses a static, 512^3 volume. While the latter algorithm runs at 30fps, ours (and our implementation of [12]) runs at 9fps, the main slow-down being the coordinate frame reparameterization (*i.e*. resampling the volume) every frame, which, while necessary, allows the sensor to move freely in world, and move into the scene to resolve more details ((c) *vs*. (g), see keyboard and plant leaves).

Our model updates almost instantly to scene changes ((d) *vs*. (h)), using $\omega = 0$, instantaneously creating a new peak in $p(v)$ at the correct disparity, but taking one more frame to surpass the magnitude of the background peak. Pathak *et al*. requires only one frame for the first peak to be largest, hence the arms (f) are more complete.

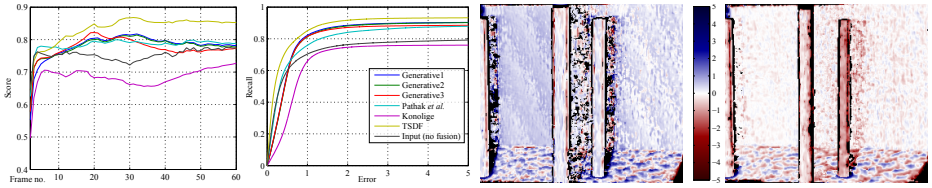
4 Conclusion

We can summarize our findings as follows: In comparison to occupancy grids which update occupancies independently *e.g*. [8], our inter-dependent visibility model produces much better depth localization. In comparison to the very similar approach of Pathak *et al*. [12], our model is theoretically more sound, and enables extensions such as our probabilistic outlier models. Overall performance was similar, but our methods had fewer occlusion boundary artifacts, and our variable outlier ratios were able to cope with a large amount of unexpected clutter.

In comparison to the TSDF [14, 15], our three methods have pros and cons. Their parameters relate directly to properties of the scene and sensor, so can be set easily, they can cope much better with outliers and react much faster to scene changes, and they provide a probabilistic certainty with the disparity estimates. Two consequences of our per-frame state reparameterization (§2.2) are that the sensor can be tracked over any trajectory, and that moving closer to objects resolves more details, in contrast to [15].

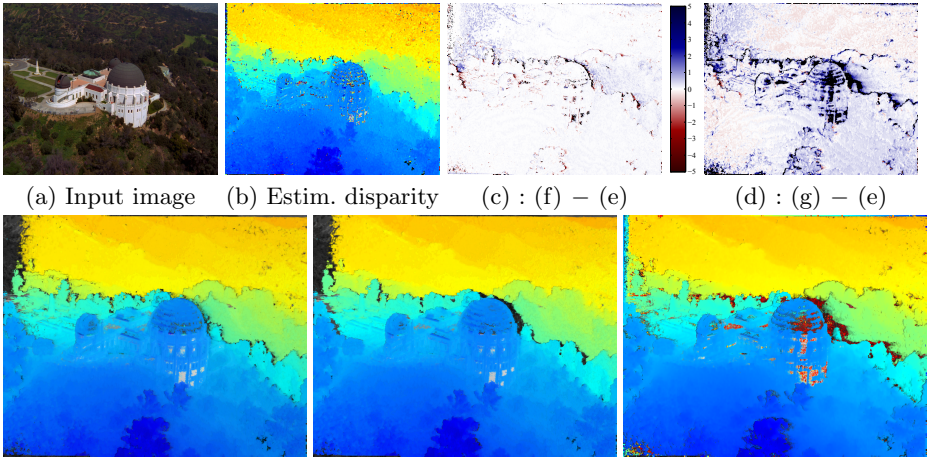
Of our three methods, Generative2 strikes a good balance between robustness to changing outlier noise conditions and avoiding too much frontal bias, as well as keeping computation and memory requirements low.

One final point, not previously acknowledged, is that all the methods tested with moving sensors have a systemic bias, TSDF's being rearward and the rest



(a) Scores (b) Error-recall (c) Generative1 err. (d) TSDF err.

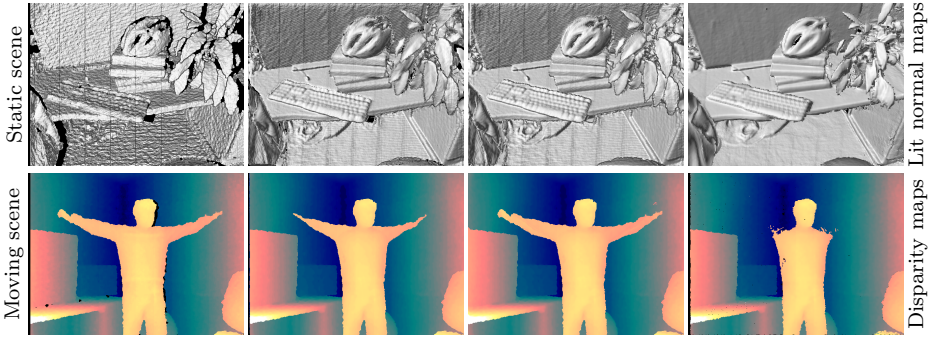
Fig. 7. Quantitative stereo. (a) Scores over time for fused stereo data from a moving camera. (b) Error-recall curves for frame 30 of the sequence. Disparity errors for frame 30 of the sequence for (c) Generative1 and (d) TSDF.



(a) Input image (b) Estim. disparity (c) : (f) − (e) (d) : (g) − (e)

(e) Generative2 (f) Pathak *et al.* (g) TSDF

Fig. 8. Qualitative stereo. (a) Frame 121 of 477 from an aerial movie. (b) The estimated disparity of the input frame. The fused disparities, weighted by certainty (TSDF weight estimated by gradient at zero crossing) and overlaid on the grayscale input frame, are shown for (e) Generative2, (f) Pathak *et al.* and (g) TSDF, with difference images (c), (d).



(a), (b) Sensor data (c), (d) Generative1 (e), (f) Pathak *et al.* (g), (h) TSDF

Fig. 9. KinectFusion. *Top row*: Lit normal maps of the geometry input (a) and computed using Generative1 (c), Pathak *et al.* (e) and TSDF (g), for a Kinect sensor moving around a static scene. *Bottom row*: Disparity maps input (b) and computed using Generative (d), Pathak *et al.* (f) and TSDF (h), for a static sensor viewing a person moving their arms up and down.

being frontal. Overcoming this bias is a key future challenge to achieving accurate online depth fusion.

Acknowledgments. We thank Björn Stenger for providing valuable feedback on this work.

References

1. Szeliski, R.: A multi-view approach to motion and stereo. In: Proceedings of CVPR. (1999)
2. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: Proceedings of ICCV. (2001)
3. Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.M., Yang, R., Nistér, D., Pollefeys, M.: Real-time visibility-based fusion of depth maps. In: Proceedings of ICCV. (2007)
4. Seitz, M.S., Dyer, C.R.: Photorealistic scene reconstruction by voxel coloring. In: Proceedings of CVPR. (1997)
5. Hernández, C., Vogiatzis, G., Cipolla, R.: Probabilistic visibility for multi-view stereo. In: Proceedings of CVPR. (2007)
6. Liu, S., Cooper, D.B.: A complete statistical inverse ray tracing approach to multi-view stereo. In: Proceedings of CVPR. (2011) 913–920
7. Elfes, A., Matthies, L.: Sensor integration for robot navigation: Combining sonar and stereo range data in a grid-based representation. In: Proceedings of IEEE Conference on Decision and Control. (1987)
8. Konolige, K.: Improved occupancy grids for map building. Autonomous Robots **4** (1997) 351–367
9. Collins, T., Collins, J.J., Ryan, C.: Occupancy grid mapping: An empirical evaluation. In: Proceedings of the Mediterranean Conference on Control Automation. (2007)
10. Guan, L., Franco, J.S., Pollefeys, M.: 3D object reconstruction with heterogeneous sensor data. In: Proceedings of 3DPVT. (2008)
11. Kim, Y.M., Theobalt, C., Diebel, J., Kosecka, J., Miscusik, B., Thrun, S.: Multi-view image and ToF sensor fusion for dense 3D reconstruction. In: Proceedings of ICCV Workshops. (2009)
12. Pathak, K., Birk, A., Poppinga, J., Schwertfeger, S.: 3D forward sensor modeling and application to grid based sensor fusion. In: Proceedings of IROS. (2007)
13. Thrun, S.: Learning occupancy grids with forward models. Autonomous Robots **15** (2001) 111–127
14. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of SIGGRAPH. (1996)
15. Newcombe, R.A., Izadi, S., Hiliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking. In: Proceedings of ISMAR. (2011)
16. Zach, C., Pock, T., Bischof, H.: A globally optimal algorithm for robust TV-L^1 range image integration. In: Proceedings of ICCV. (2007)
17. Vogiatzis, G., Hernández, C.: Video-based, real-time multi view stereo. Image and Vision Computing **29**(7) (2011) 434–441
18. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
19. Minka, T.: A family of algorithms for approximate Bayesian inference. PhD thesis, MIT (2001)
20. Point Cloud Library: http://pointclouds.org