A Closed-form Bayesian Fusion Equation using Occupancy Probabilities

Charles Loop
Microsoft Research

Qin Cai
Microsoft Research

Sergio Orts-Escolano
University of Alicante

Philip A. Chou
Microsoft Research

Abstract

We present a new mathematical framework for multi-view surface reconstruction from a set of calibrated color and depth images. We estimate the occupancy probability of points in space along sight rays, and combine these estimates using a normalized product derived from Bayes' rule. The advantage of this approach is that the free space constraint is a natural consequence of the formulation, and not a separate logical operation. We present a single closed form implicit expression for the reconstructed surface in terms of the image data and camera projections, making analytic properties such as surface normals not only easy to compute, but exact. This expression can be efficiently evaluated on the GPU, making it ideal for high performance real-time applications, such as live human body capture for immersive telepresence.

1. Introduction

Real-time surface reconstruction from multiple color and depth inputs is an important problem for the creation of telepresence in the emerging fields of Augmented (AR and Virtual (VR) Reality. Devices like Microsoft Kinect and Intel RealSense demonstrate that real-time depth inputs are available today in commodity form. However, the available techniques for combining depth inputs into a 3D geometric model are limited by the demands of a real-time system. Potential solutions have appeared in both the computer graphics and robotics literature.

In computer graphics, a weighted sum of *Truncated Sign Distance Functions* (TSDFs) [4] can be used to find an implicit function whose zero crossing is the desired surface. However, this implicit function is only defined in a narrow band containing the surface. In order to fill holes created by occlusions and to insure that the algorithm produces a closed surface, set theoretic logic is used to classify space away from this narrow band as *empty* or *unseen*. Due do this separation of *blending* and *carving*, the TSDF approach is logically more complex than the method we present here, which combines these operations into a single equation.

In Robotics, occupancy grids [5] can be used to estimate the occupancy probability of cells within a voxel grid. Cells whose occupancy probability are less than $\frac{1}{2}$ are considered to be *free space*, and thus a valid region for a Robot's path to pass through. The occupancy grid technique has been limited to voxelizations, and has not been used to extract an analytic surface. One of the difficulties in doing so has been a precise and consistent definition of the surface boundary, which has not been needed for discrete approximations.

In this paper, we resolve this theoretical aspect of the probabilistic approach used by occupancy grids and develop a new surface reconstruction algorithm that naturally combines blending and carving. Contributions of the paper include the following:

- A simple implicit function, defined by the input depth images and camera data, over all of space with analytic properties
- A free space constraint that is a natural consequence of our formulation, not a separate logical operation
- A numerically stable and efficient piecewise cubic profile curve with compact support

This paper is organized as follows. Previous work is discussed in Section 2. A detailed look at the mathematics of TSDFs and occupancy grids, followed by an overview of our algorithm, is given in Section 3. The theoretical foundations of our approach are presented in Section 4, fundamental equations in Section 5, details of our implementation in Section 6, results in Section 7, and concluding remarks in Section 8.

2. Previous Work

Our work is closely related to surface reconstruction using a sum of weighted TSDFs [11, 4]. Recently, Kinect Fusion [14] has adapted this approach to a single sensor static scene problem, where a hand held depth camera is used to scan a static object or environment. For each newly capture depth frame, the current pose relative to the initial pose is determined using a dense Iterative Closest Point algorithm [1]. As more and more depth images are acquired, the new TSDFs are integrated into the model using an incremental update formulation, providing live feedback to the user. We

978-1-5090-5407-7/16 $31.00 © 2016 IEEE
DOI 10.1109/3DV.2016.47

also provide a similar incremental update formula needed for integrating new observations into the reconstruction.

Probabilistic frameworks for surface reconstruction from depth images include maximum likelihood (ML) and maximum a posteriori (MAP) estimation of the surface given the depth image measurements [25, 24]. ML finds the surface that maximizes the likelihood of the measurement data (which is assumed Gaussian), while MAP biases the likelihood of the measurements by a surface prior that favors smooth surfaces to find the most probable surface given the data.

Bayesian methods for estimating the probability that a given volume of space is occupied have also appeared. These methods, known as *occupancy grids*, were originally used in the context of real time robot navigation [21, 5, 17, 22, 12], but were later adapted to 3D modeling [9, 10, 7, 13, 27, 19, 23, 26]. While the Bayesian methods are similar to ours, none of these works adequately considers the precise boundary between occupied and unoccupied space necessary for accurate surface reconstruction.

Another method of fusing depth images uses Poisson surface reconstruction [15, 16]. Pixels and normals from the depth images are used to construct an oriented point cloud. An implicit function whose zero level set approximates this data is found by solving the Poisson equation. The advantage of this approach is that the result is a closed surface. However, the amount of computation needed to solve for the surface precludes real-time applications. Off-line reconstruction approaches using Poisson surface reconstruction have shown impressive results [3]. Another off-line approach to surface reconstruction that considers the sampling scale of oriented point cloud data uses bases functions aligned with the normals [6].

3. Mathematical Overview

In this section, we review the key mathematical aspects of TSDFs and Occupancy Grids, highlighting the weaknesses that we address. We then present an overview our new Bayesian Fusion algorithm to illustrate our proposed improvements.

3.1. Truncated Sign Distance Functions

The *distance* used in TSDFs refers to the signed distance from a point x along a sight ray until the surface is hit. The TSDF value $d_i(x)$ along all sight rays r_i through x (e.g. a voxel center) emanating from the centers of projection of a collection of images are averaged to find a single TSDF value $D(x)$. Each $d_i(x)$ is weighted by $w_i(x)$, the cosine of the angle between the sight ray r_i and the surface normal at the point where the ray intersects the surface, yielding the formula

$$D(x) = \frac{\sum w_i(x) d_i(x)}{\sum w_i(x)}. \tag{1}$$

The distances are truncated so that observations on opposite sides of an object do not interfere with each other. Sensor noise is modeled by the maximum sensor uncertainty value and corresponds to the width of the truncation region, as shown in Figure1a.

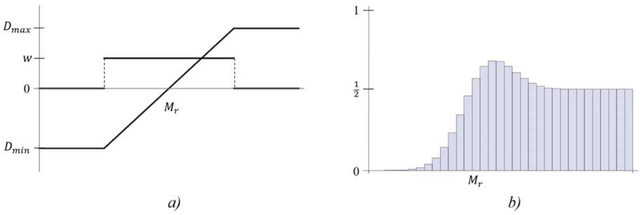

Figure 1. Plots of a) a truncated signed distance function, and b) an occupancy probability profile curve. Both are functions of the distance along a sight ray, with M_r being the measured distance to the surface.

In order to obtain a closed (watertight) surface, and to fill *holes* resulting from occluded portions of the surface, a *free space* constraint appears in the TSDF method. In practice, this constraint is crucial for removing the influence of outlier depth values that are common in real-time depth estimation. The states *unseen*, *empty*, and *near surface* are encoded using different combinations of values for $D(x)$, and $W(x) = \sum w_i(x)$ as follows:

unseen	if $D(x) = D_{max}, W(x) = 0,$
empty	if $D(x) = D_{min}, W(x) = 0,$
near surface	if $D(x) \in (D_{min}, D_{max}), W(x) > 0.$

Points are initialized to the state of *unseen*. To classify space as *empty*, distant pixels need to be identified as such, even if their distance are beyond the range of the depth sensor. One way of achieving this is using a segmentation mask and treating background pixels as far away. Points classified as *near surface* are those that lie within the truncation region. Classifying all points within a volume, rather than just the zero contour of near surface points, allows for a more general definition of what is considered to be the reconstructed surface; i.e., the boundary between unseen and empty space.

The states encoded in the values $D(x)$ and $W(x)$ are not solely the result of a combination of weighted TSDF values in (1). Rather, these values are determined by free space logic. That is, if a point is empty according to a single view, then it should be considered *empty*, even if this is contradicted by other views. This outcome cannot happen with the combining formula (1). While a single view may encode the empty state, when it is averaged with other (non-empty) views, the empty state will not be preserved. Therefore, it is not possible to use the TSDF method to express the surface as single closed-form equation.

3.2. Occupancy Grids

There are a number of formulations of occupancy grids, such as [10, 19, 23, 26], relying on Markov Random Field models or iterative Expectation-Maximization (EM) or graph cut algorithms, which at the present time are far too complex for real time use. In this section, we focus on [27], which may be the best prior work on occupancy grids suitable for real-time performance. Their work is closely related to the earlier work of [22], but handles outliers probabilistically. In the following description, for simplicity we omit the outlier handling.

An initial 3D occupancy grid is defined to be axis-aligned to a depth sensor as follows. Each pixel of the depth sensor corresponds to a ray intersecting the true surface at distance v but returning measured distance y according to a known density $p(y|v)$. Each ray is discretized into N bins. Each bin $i = 1, \ldots, N$ contains a (possibly hidden) surface or not, as indicated by a binary occupancy state variable $x_i \in \{0, 1\}$. The probability that bin i is occupied is known as the *occupancy probability*.

The posterior occupancy probabilities are stored in the N occupancy cells corresponding to the pixel of the depth sensor (see Figure 1b). These become the prior occupancy probabilities when the next set of depth measurements are integrated using Bayes' rule. However, the occupancy grid must first be re-sampled to align the grid to the next depth sensor. When all depth sensor measurements are integrated, the location of the surface is found as the bin i maximizing the posterior occupancy probability. Locating the surface by maximizing the posterior occupancy probability is an ill-posed procedure, because in higher dimensions, the locus of maxima from one direction may not coincide with that from another direction. This problem, determining the location of the "maximum" of a ridge, is why the occupancy grid framework has had difficulty in the past for 3D surface reconstruction.

3.3. Proposed Algorithm Overview

The input to our algorithm is a set of k calibrated RGBD images. For each image we have camera intrinsics and extrinsics with respect to a global coordinate system; and for each image pixel we have a color, a binary segmentation mask, and a depth estimate. The segmentation mask labels foreground and background pixels. Depth values contain sensor noise, and are set to zero when an estimate cannot be found.

For any point x in space, we estimate its occupancy probability $o_i(x) \in [0, 1]$ with respect to each image i. We then combine these estimates into a single cumulative value $O_k(x)$ for all images. If $O_k(x) < \frac{1}{2}$ then x is in empty space, if $O_k(x) > \frac{1}{2}$ then x is inside of a solid object. In the case where $O_k(x) = \frac{1}{2}$ then either x is unseen, or if $\nabla O_k(x) \neq 0$ then x is on the surface.

To estimate $o_i(x)$, we project x into the coordinate system of camera i; that is $x_i = C_i(x)$. We denote the coordinates of x_i by $\{x_{i,X}, x_{i,Y}, x_{i,Z}\}$. We use the pixel coordinates $\{x_{i,X}/x_{i,Z}, x_{i,Y}/x_{i,Z}\}$ to look up the depth d_i and assign a depth uncertainty value σ_i. We estimate $o_i(x)$ by evaluating a piecewise cubic polynomial *profile curve* as in

$$o_i(x) = H\left(\frac{x_{i,Z} - d_i}{\sigma_i}\right),$$

defined in the next section by Equation (14). In the case that x does not project to a valid image pixel, we set $o_i(x)$ to $\frac{1}{2}$.

We combine the occupancy probability estimates $o_i(x)$ for all images using (15), an incremental version of the formula

$$O_k(x) = \frac{\prod_{i=1}^{k} o_i(x)}{\prod_{i=1}^{k} o_i(x) + \prod_{i=1}^{k}(1 - o_i(x))}, \tag{2}$$

where $O_k(x)$ is a scalar valued function of points in space. The reconstructed surface is found analytically by the implicit function $O_k(x) = \frac{1}{2}$. This means we can find surface normals and other analytic properties of the surface directly. To our knowledge, no other surface reconstruction algorithm possesses this property. Furthermore, the *free space* constraint is a natural consequence of this formulation, since for a single image if $o_i(x) = 0$, then $O_k(x) = 0$ for all images.

In contrast, the TSDF method enforces a free space constraint as an additional logic layer, overriding the TSDF value when it is clear a point must be in free space. Therefore, a closed-form analytic expression for a TSDF surface is not possible.

4. The Fusion Framework

We now present the mathematical framework of our surface reconstruction approach.

Let $S \subseteq \mathbb{R}^3$ be a random set. Let $x \in \mathbb{R}^3$ be a point. Let

$$S_x = 1_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{if } x \notin S \end{cases}$$

be the *occupancy* at point x. Let r be a ray in $\mathbb{R}^3$ from the center of a depth camera through one of its pixels. Let $\mu_r(S)$ be the distance along the ray before hitting a point in S, i.e., the ground truth depth of S along r. A depth sensor measures a depth value M_r for the pixel. In such cases, $N_r = M_r - \mu_r$ is the error in the depth measurement, often assumed to be iid Gaussian with mean 0 and variance σ_r^2. In practice we could have multiple depth cameras, and all their depth measurements are denoted as $M = \{M_r\}$. Let $p(M|S)$, $P(S)$, $P(S|M)$, and $p(M)$ denote various conditional and marginal densities of M and S, using lower case p for continuous densities and upper case P for discrete densities.

In this paper, we focus on a Bayesian method known as marginal MAP (MMAP) – also known as maximum posterior marginal (MPM) or maximum marginal (MM) – estimation,

$$\hat{S}_x = \arg \max_{s \in \{0,1\}} P(S_x = s|M), \tag{3}$$

in which MAP estimation is performed point-wise for each $x \in \mathbb{R}^3$. Since the maximum is performed for each x over only two possible values, and hence is well suited for real time processing on a GPU.

4.1. Computation of the Occupancy Probability

To compute the posterior probability of occupancy $P(S_x = 1|M)$ in (3), we may again use Bayes' Theorem,

$$P(S_x = 1|M) = \frac{p(M|S_x = 1)P(S_x = 1)}{\sum_{s=0}^{1} p(M|S_x = s)P(S_x = s)}. \tag{4}$$

Hence we need to define both $p(M|S_x = s)$ and $P(S_x = s)$ for $s = 0, 1$. To define the latter, we assume the minimally informative prior on S, in which the set of variables $\{S_x\}$ are iid and

$$P(S_x = s) = \frac{1}{2} \tag{5}$$

for $s = 0, 1$. To define the former, we note that since the $\{S_x\}$ are iid, the set of measurements $M_x \triangleq \{M_r : r \ni x\}$ whose rays pass through x are conditionally independent given S_x (because x is the only point they have in common), and moreover all the measurements $M_{\bar{x}} \triangleq \{M_r : r \not\ni x\}$ whose rays do not pass through x are (primarily) dependent on S_x only through M_x. Hence

$$\begin{aligned} p(M|S_x = s) &= p(M_{\bar{x}}|M_x, S_x = s)p(M_x|S_x = s) \\ &\approx p(M_{\bar{x}}|M_x)p(M_x|S_x = s) \\ &= p(M_{\bar{x}}|M_x) \prod_{r \ni x} p(M_r|S_x = s). \end{aligned} \tag{6}$$

Combining (4), (5), and (6), we obtain

$$P(S_x = 1|M) = \frac{\prod_{r \ni x} p(M_r|S_x = 1)}{\sum_{s=0}^{1} \prod_{r \ni x} p(M_r|S_x = s)}. \tag{7}$$

To simplify (7), we rewrite $p(M_r|S_x = s)$ again using Bayes' Theorem:

$$p(M_r|S_x = s) = P(S_x = s|M_r)p(M_r)/P(S_x = s), \tag{8}$$

whence (7), using (5), becomes

$$P(S_x = 1|M) = \frac{\prod_{r \ni x} P(S_x = 1|M_r)}{\sum_{s=0}^{1} \prod_{r \ni x} P(S_x = s|M_r)}. \tag{9}$$

4.2. Single Ray Behavior

We now consider how to compute the occupancy probability $P(S_x = 1|M_r)$ at a point x given a depth measurement along a single ray r originating at a camera center and passing through point x, as seen in Figure 2. Here, d_{xr} is the distance to x along the ray, μ_r is the true distance to the surface, and M_r is the measured distance to the surface. Remembering that S is random and thus so is μ_r, we see that $S \to \mu_r \to M_r$ is a Markov chain, and thus so is $M_r \to \mu_r \to S$. Hence $P(S_x|\mu_r, M_r) = P(S_x|\mu_r)$ and the occupancy probability $P(S_x = 1|M_r)$ is given by:

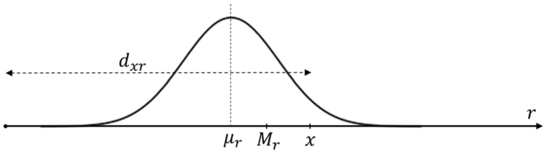

Figure 2. Measurement distribution along a particular ray r.

$$P(S_x = 1|M_r) = \int_0^{\infty} P(S_x = 1|\mu_r)\, p(\mu_r|M_r)\, d\mu_r. \tag{10}$$

To evaluate this, first we must select models for the probability of occupancy given the true depth $P(S_x = 1|\mu_r)$ and the distribution of the true depth given a noisy measurement $p(\mu_r|M_r)$.

4.2.1 Probability of Occupancy given True Depth

For $P(S_x = 1|\mu_r)$, clearly the point x must be unoccupied if it is in front of the surface, it must be occupied if it is just within the surface, and it may or may not be occupied if it is behind the surface. Intuitively, however, if x is not too far behind the surface, it is more likely to be occupied, as real-world objects tend to have a certain minimum thickness, say τ. Thus we model the probability of occupancy given the true depth as

$$P(S_x = 1|\mu_r) = \begin{cases} 0 & \text{if } d_{xr} < \mu_r, \\ 1 & \text{if } \mu_r \leq d_{xr} < (\mu_r + \tau), \\ \frac{1}{2} & \text{if } (\mu_r + \tau) < d_{xr}, \end{cases} \tag{11}$$

as illustrated in Figure 3a.

4.2.2 Distribution of True Depth given Measurement

Next we turn to the model for $p(\mu_r|M_r)$, the distribution of the true depth given a noisy measurement. Typically, measurement noise is modeled by a Gaussian

$$p(M_r|\mu_r) = \frac{1}{\sqrt{2\pi\sigma_r^2}} e^{-\frac{(M_r - \mu_r)^2}{2\sigma_r^2}}$$

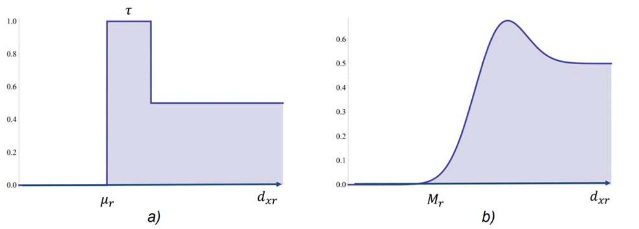

Figure 3. a) A graphical depiction of $P\left(S_x = 1|\mu_r\right)$, the profile curve given truth depth. b) A graphical depiction of the profile curve $p\left(S_x = 1|M_r\right) = \int_0^\infty P\left(S_x = 1|\mu_r\right) p\left(\mu_r|M_r\right) d\mu_r$ using a Gaussian noise model.

with mean μ_r and variance σ_r^2. Since

$$p(\mu_r|M_r) = \frac{p(M_r|\mu_r)p(\mu_r)}{p(M_r)},$$

if μ_r has a uniform prior, then so will M_r, and $p(\mu_r|M_r)$ will have the same functional form as $p(M_r|\mu_r)$, namely

$$p\left(\mu_r|M_r\right) = \frac{1}{\sqrt{2\pi\sigma_r^2}} e^{-\frac{(\mu_r - M_r)^2}{2\sigma_r^2}}. \tag{12}$$

In sum, evaluating (10) using (11) and (12) we obtain

$$\begin{aligned} P\left(S_x = 1|M_r\right) = & \frac{1}{2}\text{Erf}\left(\frac{M_r - d_{xr}}{\sqrt{2}\sigma_r}\right) \\ & - \frac{1}{4}\text{Erf}\left(\frac{M_r - d_{xr} - \tau}{\sqrt{2}\sigma_r}\right) + \frac{1}{4}, \end{aligned} \tag{13}$$

where $\text{Erf}\left(x\right) = \frac{2}{\pi}\int_0^x e^{-t^2} dt$ is the well known *error function*. We refer to $P(S_x = 1|M_r)$ as a *profile curve*, noting its prominent role in our reconstruction algorithm. A plot of this particular profile curve appears in Figure 3b.

4.2.3 The Problem with Gaussian Measurement Noise

According to the MMAP objective, the estimated surface is located at the point where the occupancy probability transitions from less than $\frac{1}{2}$ to greater than $\frac{1}{2}$. However, it can be seen from (13) that even when the measurement is equal to the true depth $M_r = \mu_r$, the occupancy probability at the true depth $d_{xr} = \mu_r$ is strictly less than $\frac{1}{2}$. In practical terms, this means that the estimated surface lies slightly behind the depth measurement even when it is accurate. In short, the estimate is inconsistent. The reason is that the support of the Gaussian is infinite while τ is finite.

One way to fix the problem is to set $\tau = \infty$. Then, the profile curve $P(S_x = 1|M_r = \mu_r) = \frac{1}{2}$ exactly when $d_{xr} = \mu_r$. Unfortunately, in this case, the occupancy probabilities may degenerate when combined. This problem arises when the occupancy probability along two distinct rays intersecting at a point x are close to 0 and 1 respectively. In the first case, x is clearly in front of a surface, so $P(S_x = 1|M_{r_1}) \approx 0$. In the second case x is occluded, hence is behind a surface, so $P(S_x = 1|M_{r_2}) \approx 1$. In this case, normalization will fail since the denominator in (9) will vanish.

We instead choose to limit the support of measurement noise, which in practice does not have unbounded support. One way would be to truncate the Gaussian noise model. However, this is problematic as the resulting curve $P(S_x|M_r)$ would have discontinuities.

4.3. Quadratic B-Spline Measurement Noise

To overcome these difficulties, we move to a quadratic B-spline noise model $p(\mu_r|M_r) = Q\left(\frac{\mu_r - M_r}{\sigma_r}\right)$, where

$$Q\left(t\right) = \begin{cases} \frac{1}{16}\left(3+t\right)^2 & \text{if } -3 \leq t \leq -1, \\ \frac{1}{8}\left(3-t^2\right) & \text{if } -1 < t < 1, \\ \frac{1}{16}\left(3-t\right)^2 & \text{if } 1 \leq t \leq 3, \\ 0 & \text{otherwise.} \end{cases}$$

We compare this curve to a Gaussian in Figure 4. Note that this quadratic B-spline basis function also has unit integral. Unlike a Gaussian whose support is infinite, $Q\left(\frac{\mu_r - M_r}{\sigma_r}\right)$ is non-zero over $6\sigma_r$ units; $3\sigma_r$ to the left and right of M_r. By choosing our thickness term $\tau = 3\sigma_r$, then $\int_{\mu_r=0}^\infty P(S_x|\mu_r)Q\left(\frac{\mu_r - M_r}{\sigma_r}\right) d\mu_r = \frac{1}{2}$, since exactly half the area under the curve $Q\left(\frac{\mu_r - M_r}{\sigma_r}\right)$ is under the unit pulse part of $P(S_x|\mu_r)$, the other half is zeroed out, as shown in the left half of Figure 5. The integral is solved by the difference

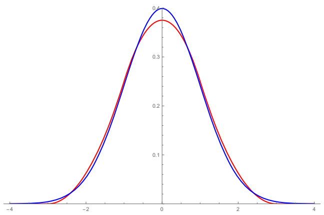

Figure 4. Our quadratic B-Spline noise function (red), compared to a Gaussian (blue).

$$\boxed{H(t) = Q_{cdf}\left(t\right) - \frac{1}{2}Q_{cdf}\left(t-3\right)} \tag{14}$$

where

$$Q_{cdf}\left(t\right) = \begin{cases} 0 & \text{if } t < -3, \\ \frac{1}{48}\left(3+t\right)^3 & \text{if } -3 \leq t \leq -1, \\ \frac{1}{2} + \frac{1}{24}t\left(3+t\right)\left(3-t\right) & \text{if } -1 < t < 1, \\ 1 - \frac{1}{48}\left(3-t\right)^3 & \text{if } 1 \leq t \leq 3, \\ 1 & \text{if } 3 < t, \end{cases}$$

is the cumulative density of the quadratic B-Spline noise function $Q(t)$. A plot of $H(t)$ appears in the right half of

Figure 5. Note that $H(t)$ is a C^2 continuous piecewise cubic curve. Also note that the support of $H(t)$ is limited to $-3 < t < 6$, outside this range $H(t)$ can only take on the values 0 or $\frac{1}{2}$. Formally then, we have

$$\begin{aligned} P(S_x = 1|M_r) &= \int_{\mu_r=0}^{\infty} P(S_x = 1|\mu_r) Q\left(\frac{\mu_r - M_r}{\sigma_r}\right) d\mu_r \\ &\triangleq H\frac{d_{xr} - M_r}{\sigma_r}. \end{aligned}$$

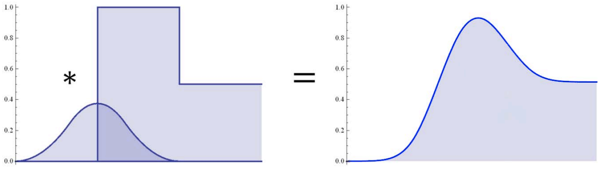

Figure 5. On the left, we show the geometric construction of $P(S_x|\mu_r)$, where the parameter τ equals half the support a quadratic B-spline basis function used for $p(\mu_r|M_r)$. On the right , we show the resulting convolution $\int_{\mu_r=0}^{\infty} P(S_x = 1|\mu_r)Q\left(\frac{\mu_r-M_r}{\sigma_r}\right) d\mu_r$ as the piecewise cubic profile curve $H(t)$.

Note that for an accurate measurement $M_r = \mu_r$, the occupancy probability $P(S_x = 1|M_r = \mu_r)$ at the true depth $d_{xr} = \mu_r$ is exactly $\frac{1}{2}$. Moreover, the occupancy probability stays at $\frac{1}{2}$ as more accurate measurements are combined using (7). Indeed, as more accurate measurements along the same ray are combined, the occupancy probability goes to 0 for $d_{xr} < \mu_r$, goes to 1 for $d_{xr} \in (\mu_r, \mu_r + \tau)$, and stays at $\frac{1}{2}$ for $d_{xr} = \mu_r$ and $d_{xr} \geq \mu_r + \tau$. In short, the estimate is consistent. We achieve this by equating nominal surface thickness and sensor noise. We believe this is reasonable, as surface thickness is an arbitrary parameter.

5. Fundamental Equations

We now define our fundamental closed-form equations for the surface. In order to streamline our notation, we make the substitutions

$$\begin{aligned} P(S_x = 1|M_{r_1}, \ldots, M_{r_k}) &\rightarrow O_k(x), \\ P(S_x = 1|M_{r_i}) &\rightarrow o_i(x). \end{aligned}$$

Conceptually, $O_k(x)$ is the occupancy probability of x given several measurements $M_{r_1}, \ldots, M_{r_k}$, while $o_i(x)$ is the occupancy probability of x given the single measurement M_{r_i}.

Rewriting (9), we get the occupancy probability at the point x, described in Equation (2). Equation (2) shows that the occupancy probability at point x is the normalized product of the occupancy probability of the individual measurement rays passing through x. As more and more depth measurements are taken, the number of rays passing x may increase. Numerically, this may be unstable as asymptotically, the products in (2) will approach zero, since occupancy probabilities are typically less than one. Fortunately, an *incremental update formula* is available for combining new observations into a cumulative total of the occupancy probability at point x, as shown in the following:

CLAIM:

$$\boxed{O_k(x) = \frac{O_{k-1}(x)o_k(x)}{O_{k-1}(x)o_k(x) + (1 - O_{k-1}(x))(1 - o_k(x))}} \tag{15}$$

where $O_0(x) = \frac{1}{2}$.

PROOF : For $k > 1$, decrement k by 1 in (2) and substitute into (15), then simplify. For $k = 1$, we have $O_1(x) = o_1(x)$. □

The reconstructed surface is completely defined by the implicit form

$$\boxed{O_k(x) = \frac{1}{2}}$$

Note that we can analytically express derivatives and other properties of the surface using these equations. In order to avoid considering unseen regions of space where $O_k(x) = \frac{1}{2}$ as a part of the surface, we add the restriction that $\nabla O(x) \neq 0$.

In the next section we discuss how we use this to implement our surface reconstruction algorithm.

6. Implementation

The scope of this paper is limited to our theoretical contributions. It is not our intention to describe our complete system here. Like many other surface reconstruction systems we use voxels, and evaluate occupancy probabilities at voxel vertices. We use specialized acceleration structures and sparse octrees to realize an effective voxel resolution of 1024^3. In practice, we compute billions of vertex occupancy probabilities per second to achieve real-time performance.

We now describe these calculations in detail. Our input is a set of depth images $I_i : \mathbb{R}^2 \rightarrow \mathbb{R}$ and a corresponding set of calibrated cameras $C_i : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. (RGB images do not influence the geometry of our reconstruction, only shading.) We denote the components of C_i by

$$C_i(x) = \left[\begin{array}{ccc} c_{i,X}(x) & c_{i,Y}(x) & c_{i,Z}(x) \end{array}\right].$$

Thus

$$I_i\left(\frac{c_{i,X}(x)}{c_{i,Z}(x)}, \frac{c_{i,Y}(x)}{c_{i,Z}(x)}\right) \rightarrow M_{r_i}, \quad \text{and} \quad c_{i,Z}(x) \rightarrow d_{xr_i}.$$

In practice, we model sensor noise as the expected depth error from stereo, which can be shown to be

$$\kappa \, c_{i,Z}(x)^2 \rightarrow \sigma_{r_i},$$

where κ depends on the baseline, focal length, and matching error of the stereo system [8]. With these assignments, we evaluate the profile curve (Equation (14)) for each ray r_i passing through x to get the occupancy probabilities

$$\boxed{o_i(x) \;=\; H\left(\frac{c_{i,Z}(x) - I_i\left(\frac{c_{i,X}(x)}{c_{i,Z}(x)}, \frac{c_{i,Y}(x)}{c_{i,Z}(x)}\right)}{\kappa \, c_{i,Z}(x)^2}\right)} \qquad (16)$$

Note that if the point x projects outside image I_i under camera C_i, we set the occupancy probability to $\frac{1}{2}$ for that view, since we obtain no information about the occupancy of x in this case. As each single view occupancy probability is evaluated, we use (15) to find the combined total.

In our system, we render voxels as splats, similar to [20]. Voxels are colored based on their projections into the RGB images, similar to the approach in [18].

7. Results

We evaluated our Bayesian Fusion framework on a variety of multi-view configurations with color and depth inputs. Figure 6 shows the result from four synchronized RGBD inputs. The black area of the depth map means unknown depth value due to side effects of the depth generation algorithm [2], but it's within the segmentation mask.

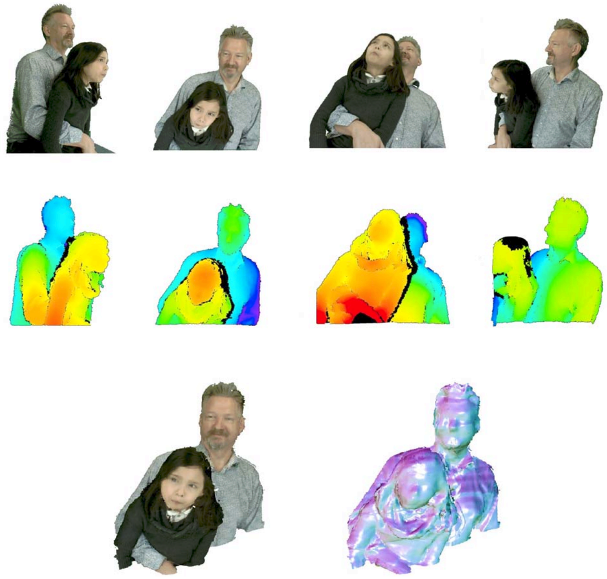

Figure 6. The top row shows the color input of our 4 views with fg/bg segmentation. The middle row shows the corresponding depth images. The Bayesian Fusion reconstruction results in one novel view are shown at the bottom, both in colored voxel and normal map.

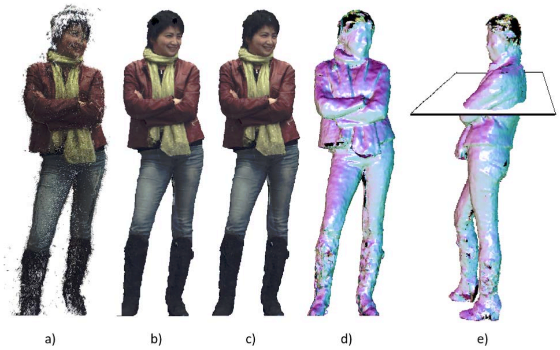

Figure 7. The reconstruction results with orthographical view from a) the point cloud merged by the color and depth inputs, b) TSDF, c) Bayesian Fusion, d) normals, e) a plane slice to get contour of the 3D form.

Next, we demonstrate surface reconstruction on full body capture with eight segmented RGBD images. The capture region is a two meter cube, which allows free full body movements. The results depicted in Figure 7 show reconstruction using orthographical projection, in color and normal maps. Figure 8 presents key frames of subjects with different poses and motion dynamics, all computed in real-time. In the examples, the bottom row with a subject twirling scarfs in the air, is challenging due to motion blur and inaccurate depth, the top row using the same data set from [3], shows two subjects dancing together, which involves many occlusions.

With a voxel resolution of 1024^3 it is hard to perceive the subtle difference of reconstruction results from Bayesian Fusion and TSDF qualitatively, see Figures 7b and 7c, compared to the raw point cloud in Figure 7a. For this comparison, we apply a horizontal plane to the 3D form, depicted in (Figure 7e). When we compare the outline contours using TSDF and our method, the difference becomes noticeable. In Figure 9, the top row shows the result from TSDF, and the bottom from Bayesian Fusion, using the data from Figure 7. Bayesian Fusion shows cleaner and smoother surfaces compared to that of TSDF.

Quantitatively, we found no measurable difference between our approach and the TSDF method in practice. To isolate the performance of profile curve evaluation, we ran three tests, one for our cubic spline profile curve (Spline), another for the Gaussian noise profile curve (Gaussian), and a third that returned truncated signed distance (TSD). We launched these CUDA kernels with 1000 blocks of 1000 threads, where each thread did 1000 evaluations over independent parameter values; in all, 1 billion evaluations per kernel. Average running times on a GeForce TitanX GPU are shown in the table below.

Kernel	Spline	Gaussian	TSD
Time (ms)	12.3	34.9	7.5

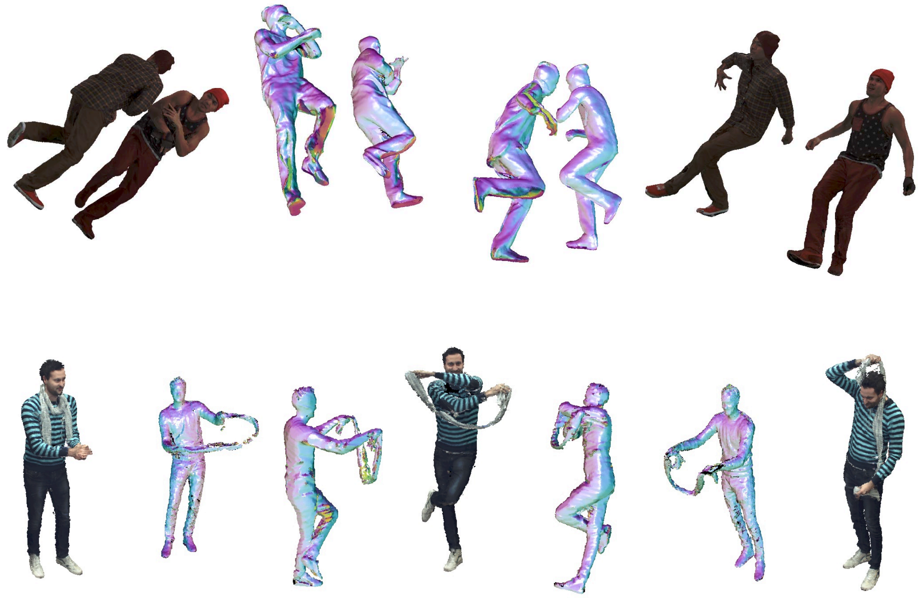

Figure 8. Multiple examples of surface reconstruction in Bayesian Fusion using eight views of RGBD inputs.

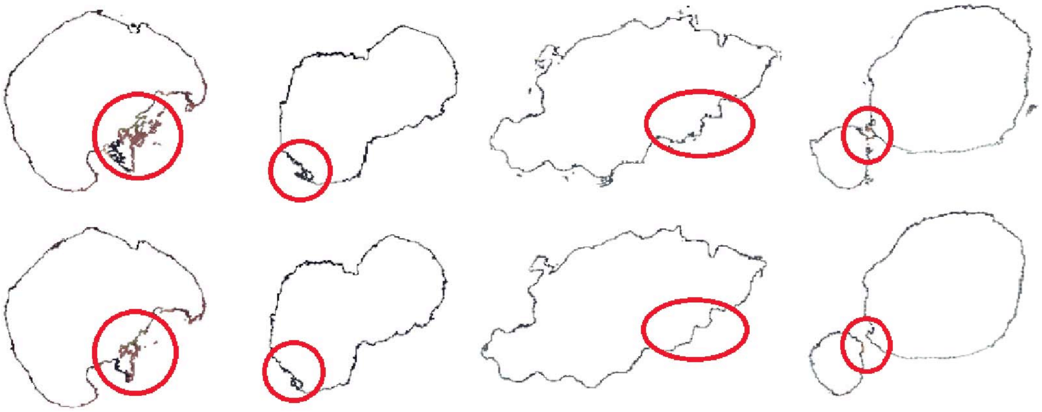

Figure 9. The top row shows the result using weighted TSDFs and the bottom row is the corresponding results from Bayesian Fusion.

Note that cubic spline profile curve evaluation is significantly faster than one based on Gaussian noise. While more costly than truncated signed distance, we point out that this test does not account for the weight evaluation of the TSDF approach, which will add overhead. Also note that 1 billion profile curve evaluations is at least 2 orders of magnitude more than would be needed for even the most complex single frame example shown here. In short, despite requiring the evaluation of a cubic profile curve, there is no meaningful performance degradation compared to TSDF's.

8. Conclusion

We have presented a new surface reconstruction formulation that naturally combines probabilistic blending and carving. This formulation is based on estimating *occupancy probabilities* along rays from a depth sensor camera center, through depth pixels. We find the occupancy probability by evaluating a cubic polynomial profile curve at a value corresponding to the signed distance to a noisy surface observation. We combine these probabilities using a blending rule derived as the solution to a MMAP problem. Since all points in space can be evaluated using only the blending rule and image/calibration data, we can write a closed-form implicit function for the reconstructed surface. We believe this function is useful for determining analytic properties of the surface. Furthermore, occupancy probability is easily related to sensor uncertainty, giving a principled way of dealing with such uncertainty in the reconstruction process.

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992. 1

[2] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *British Machine Vision Conference, BMVC 2011, Dundee, UK, August 29 - September 2, 2011. Proceedings*, 2011. 7

[3] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan. High-quality streamable free-viewpoint video. *ACM Trans. Graph.*, 34(4), 2015. 2, 7

[4] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pages 303–312, 1996. 1

[5] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, 22(6):46–57, 1989. 1, 2

[6] S. Fuhrmann and M. Goesele. Floating scale surface reconstruction. *ACM Trans. Graph.*, 33(4):46:1–46:11, 2014. 2

[7] R. Furukawa, T. Itano, A. Morisaka, and H. Kawasaki. Shape-merging and interpolation using class estimation for unseen voxels with a gpu-based efficient implementation. In *Proc. IEEE 6th International Conference on 3-D Digital Imaging and Modeling*, pages 289–296, Aug. 2007. 2

[8] D. Gallup, J. Frahm, P. Mordohai, and M. Pollefeys. Variable baseline/resolution stereo. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008. 7

[9] P. Gargallo and P. Sturm. Bayesian 3d modeling from images using multiple depth maps. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2005. 2

[10] C. Hernandez, G. Vogiatzis, and R. Cipolla. Probabilistic visibility for multi-view stereo. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2007. 2, 3

[11] A. Hilton, A. J. Stoddart, J. Illingworth, and T. Windeatt. Reliable surface reconstructiuon from multiple range images. In *Computer Vision - ECCV'96, 4th European Conference on Computer Vision, Cambridge, UK, April 15-18, 1996, Proceedings, Volume I*, pages 117–126, 1996. 1

[12] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Auton. Robots*, 34(3):189–206, 2013. 2

[13] X. Hu and P. Mordohai. Least commitment, viewpoint-based, multi-view stereo. In *Proc. Int'l Conf. 3D Imaging, Modeling, Processing, Visualization & Transmission (3DimPVT)*, Oct. 2012. 2

[14] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. A. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. J. Davison, and A. W. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, October 16-19, 2011*, pages 559–568, 2011. 1

[15] M. M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, June 26-28, 2006*, pages 61–70, 2006. 2

[16] M. M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29, 2013. 2

[17] K. Konolige. Improved occupancy grids for map building. *Auton. Robots*, 4(4):351–367, 1997. 2

[18] C. Kuster, T. Popa, C. Zach, C. Gotsman, and M. H. Gross. Freecam: A hybrid camera system for interactive free-viewpoint video. In *Proceedings of the Vision, Modeling, and Visualization Workshop 2011, Berlin, Germany, 4-6 October, 2011*, pages 17–24, 2011. 7

[19] S. Liu and D. B. Cooper. Statistical inverse ray tracing for image-based 3d modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(10):2074–2088, 2014. 2, 3

[20] C. Loop, C. Zhang, and Z. Zhang. Real-time high-resolution sparse voxelization with application to image-based modeling. In *High-Performance Graphics 2013, Anaheim, California, USA, July 19-21, 2013. Proceedings*, pages 73–80, 2013. 7

[21] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988. 2

[22] K. Pathak, A. Birk, J. Poppinga, and S. Schwertfeger. 3d forward sensor modeling and application to occupancy grid based sensor fusion. In *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, Oct. 2007. 2, 3

[23] N. Savinov, L. Ladicky, C. Hane, and M. Pollefeys. Discrete optimization of ray potentials for semantic 3d reconstruction. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2, 3

[24] W. Sun, G. Cheung, P. A. Chou, D. Florencio, C. Zhang, and O. Au. Rate-constrained 3d surface estimation from noise-corrupted multiview depth videos. *IEEE Transactions on Image Processing*, July 2014. 2

[25] W. Sun, G. Cheung, P. A. Chou, D. Florencio, C. Zhang, and O. C. Au. Rate-distortion optimized 3d reconstruction from noise-corrupted multiview depth videos. IEEE Internation Conference on Multimedia & Expo (ICME), July 2013. 2

[26] A. O. Ulusoy, A. Geiger, and M. J. Black. Towards probabilistic volumetric reconstruction using ray potentials. In *Proc. Int'l Conf. 3D Vision (3DV)*, Oct. 2015. 2, 3

[27] O. J. Woodford and G. Vogiatzis. A generative model for online depth fusion. In *Proc. European Conference of Computer Vision (ECCV)*, Oct. 2012. 2, 3