Bayesian 3D Modeling from Images using Multiple Depth Maps

Pau Gargallo and Peter Sturm

INRIA Rhône-Alpes, GRAVIR-CNRS, Montbonnot, France

Abstract

This paper addresses the problem of reconstructing the geometry and color of a Lambertian scene, given some fully calibrated images acquired with wide baselines. In order to completely model the input data, we propose to represent the scene as a set of colored depth maps, one per input image. We formulate the problem as a Bayesian MAP problem which leads to an energy minimization method. Hidden visibility variables are used to deal with occlusion, reflections and outliers. The main contributions of this work are: a prior for the visibility variables that treats the geometric occlusions; and a prior for the multiple depth maps model that smoothes and merges the depth maps while enabling discontinuities. Real world examples showing the efficiency and limitations of the approach are presented.

1. Introduction

This paper addresses the problem of recovering high-resolution 3D models of a scene from a small collection of images. Reconstruction of 3D models from images has been widely studied in computer vision. Many algorithms have been proposed. Differences between them lie in the model used to represent the scene, the prior on this model and the optimization method used for estimating it. The scene representation is a very important factor that practically determines the strengths and weaknesses of the approaches.

Volumetric models, such as voxel-based ones [8, 1, 10] or using level-sets [3], are based on a discretization of 3D space and their goal is to determine the full and the empty cells. These methods can use a large number of images taken from arbitrarily placed viewpoints. Any shape can be represented and the visibility problem is handled in a deterministic geometric manner. However, the initial discretization limits their resolution. The only way of increasing the resolution is to increase the size of the voxel grid. On the other hand, mesh representations [6, 9, 19] can, in theory, adapt their resolution to best reconstruct detailed shapes, but have problems dealing with self-intersections and topological changes during the search.

Depth maps have been mainly studied for two views with a small baseline [12, 7, 15, 18]. The small baseline makes it impossible to get accurate results and these methods are forced to use strong priors that usually introduce fronto-parallel bias. The results of these methods are not precise continuous depth maps but piece-wise planar surfaces. Recently depth map reconstruction from multiple wide-baseline images has been developed with impressive results [14, 13]. The wide-baseline configuration allows astonishingly accurate results without the discretization and topological problems of other methods.

These nice properties of the depth map representation encourage us to use it. However, a single depth map is usually not enough to represent the whole scene: only the parts viewed in a reference view are modeled. A depth map for every input image [17] is needed to ensure that every input pixel is used and modeled. This is probably the model best adapted to the resolution of the input and is the model treated in this work. Alternatively to computing each depth map independently and merging them in a postprocessing step [11, 13], we will compute all the depth maps at the same time which permits an efficient geometric visibility/occlusion reasoning and ensures that the output depth maps will be coherent.

In [13], depth map recovery was formulated as a maximum a posteriori (MAP) problem using the framework proposed in [4] for the novel view synthesis problem showing that the two problems are intrinsically the same. Here we adopt this framework and adapt it to the case of multiple reference views.

The main contributions of this paper to this framework are: First, a reflection on and modification of the likelihood formula. Second, a geometric visibility prior. We use the current depth maps estimation to determine the prior on visibility of the model points. And finally, a multiple depth maps prior that smoothes and merges the depth maps while preserving discontinuities.

1.1. Problem Statement

Our goal is to find a 3D representation of a scene, from a given set of images with full calibration information, i.e. known intrinsic and extrinsic parameters. The model we use to represent the scene consists of a set of colored depth maps. For every pixel in the input images, we want to infer the depth and color of the 3D point that this pixel is seeing.

2. Modeling and Estimation

We treat the problem as a Bayesian MAP search. Input images $\mathcal{I}$ are regarded as a noisy measurement of the model θ. The researched model is defined as the one that maximizes the posterior probability $p(\theta|\mathcal{I}) \propto p(\mathcal{I}|\theta)p(\theta)$.

We first define the relevant variables of the problem in section 2.1. Next, in section 2.2 we decompose the joint probability of the variables, determining the statistical dependencies between them. In sections 2.3 to 2.5 we give a form to each term of the decomposition. Finally in 2.6 we present the optimization method used to estimate the MAP.

2.1. Depth and Color Maps and Visibility Variables

The set of n input images is noted as $\{\mathcal{I}_i\}_{i=1..n}$. $\mathcal{I}_i(\mathbf{x})$ is the color of pixel $\mathbf{x}$ in the i^{th} image and lives in some color space (graylevel, RGB, etc.). The cameras are represented by a set of projection matrices $\{\mathrm{P}_i\}_{i=1..n}$. These matrices have the usual form $\mathrm{P}_i = \mathrm{K}_i(\mathrm{R}_i|\mathrm{t}_i)$ and we scale them so that $(\mathrm{K}_i)_{3,3} = 1$. The depth of a point $\mathbf{X} = (X, Y, Z)^\top$ with respect to a camera position P_i is then defined as $d_i(\mathbf{X}) = (\mathrm{P}_i\bar{\mathbf{X}})_3$, where $\bar{\mathbf{X}} = (X, Y, Z, 1)^\top$. Conversely, if pixel $\mathbf{x} = (x, y)^\top$ of image i has a depth d, then the euclidean coordinates of the corresponding 3D point are $\mathbf{X}_i(\mathbf{x}, d) = d(\mathrm{K}_i\mathrm{R}_i)^{-1}\bar{\mathbf{x}} - \mathrm{R}_i^\top \mathrm{t}_i$, where $\bar{\mathbf{x}} = (x, y, 1)^\top$.

For every pixel in the input images we will compute its depth and color. Depths will be stored in a set of depth maps $\{\mathcal{D}_i\}_{i=1..n}$ and colors in a set of ideal images $\{\mathcal{I}_i^*\}_{i=1..n}$. $\mathcal{D}_i(\mathbf{x})$ and $\mathcal{I}_i^*(\mathbf{x})$ will then be the depth and the color of the point seen by the pixel $\mathbf{x}$ of the i^{th} image. Sometimes it will be more illustrative to think of the set of colored depth maps as a representation of the 3D point cloud $\{\mathbf{X}_i(\mathbf{x}, \mathcal{D}_i(\mathbf{x})) : i = 1..n, \mathbf{x} \in \mathcal{I}_i\}$ and treat all the points of the cloud in the same manner, ignoring their origin, i.e. the image by whose depth map a point is parameterized.

For simplicity, given a point $\mathbf{X} = \mathbf{X}_i(\mathbf{x}, \mathcal{D}_i(\mathbf{x}))$ in the cloud, its estimated color $\mathcal{I}_i^*(\mathbf{x})$ will be noted by $C(\mathbf{X})$. The value of other images on its projection will be noted as $\mathcal{I}_j(\mathbf{X})$ instead of $\mathcal{I}_j(\mathrm{P}_j\bar{\mathbf{X}})$. Similarly, we write $\mathcal{D}_j(\mathbf{X})$ instead of $\mathcal{D}_j(\mathrm{P}_j\bar{\mathbf{X}})$. It is important to note that this is the estimated depth of the *pixel* of image $\mathcal{I}_j$, onto which the 3D point $\mathbf{X}$ is projected, and not the actual depth of the *3D point* $\mathbf{X}$ itself. The latter will be noted as $d_j(\mathbf{X})$, see above and figure 1. As $\mathbf{X}$ is parameterized by the depth map of image $\mathcal{I}_i$, of course, $\mathcal{D}_i(\mathbf{X}) = d_i(\mathbf{X})$.

Due to geometric occlusions, specular reflections or other effects not all the points of the cloud will be visible in every input image. As proposed by Strecha et al. [13] we introduce a boolean variable $\mathcal{V}_{i,\mathbf{X}}$ for each model point $\mathbf{X}$ and each image $\mathcal{I}_i$, that signals whether $\mathbf{X}$ is visible or not in image $\mathcal{I}_i$. These variables are hidden and only their probabilities will be computed.

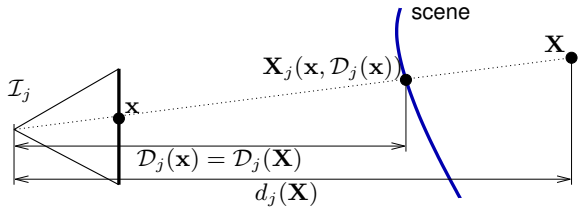

Figure 1. For a given 3D point $\mathbf{X}$, $d_j(\mathbf{X})$ denotes its depth relative to image $\mathcal{I}_j$. $\mathbf{D}_j(\mathbf{X})$ denotes the estimated depth of the pixel onto which $\mathbf{X}$ is projected by P_j, $\mathbf{x} = \mathrm{P}_j\bar{\mathbf{X}}$.

2.2. Decomposition

Having all the variables defined, the next step in a Bayesian modeling task is to choose a decomposition of their joint probability. The decomposition will define the statistical dependencies between the variables that our model is considering. For completeness, we add to the previously defined variables, a variable $\tau = \{\Sigma, \sigma, \sigma', v, l\}$, that represents the set of all the parameters that will be used in our approach, see below. The joint probability of all the variables is then $p(\mathcal{I}, \mathcal{V}, \mathcal{I}^*, \mathcal{D}, \tau)$ and the proposed decomposition (fig. 2):

$$p(\tau)\; p(\mathcal{I}^*|\tau)\; p(\mathcal{D}|\tau)\; p(\mathcal{V}|\mathcal{D}, \tau)\; p(\mathcal{I}|\mathcal{V}, \mathcal{I}^*, \mathcal{D}, \tau) \qquad (1)$$

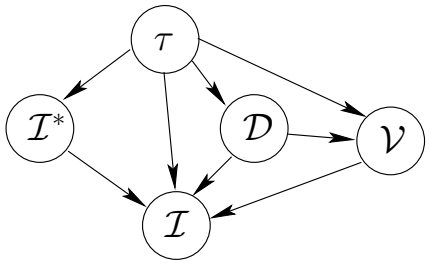

Figure 2. Network representation of the joint probability decomposition. Arrows represent statistical dependencies between variables.

1. $p(\tau)$ is the prior probability of the parameters. We assume a uniform one in this work and ignore this term.
2. $p(\mathcal{I}^*|\tau)$ is the prior on the colors of the depth maps. This term was used by Fitzgibbon et al. [4] to regularize the novel view synthesis problem with great success. The so-called image-based priors were introduced to enforce the computed images $\mathcal{I}^*$ to look like natural images, which in practice was enforced by looking like images of a catalogue of examples [5]. In this work, we adopt a uniform prior, centering the regularization on the depth maps like [13].
3. $p(\mathcal{D}|\tau)$ is the prior on depth maps. Its work is to smooth and integrate the different depth maps. It is developed in section 2.5. Note that in contrast with [13], no statistical dependence between $\mathcal{I}^*$ and $\mathcal{D}$ is used here. Modeling this dependence can help when dealing with constant

albedo surfaces were image and depth discontinuities are correlated. On the other hand, this can produce undersmoothing of textured smooth surfaces.

4. $p(\mathcal{V}|\mathcal{D},\tau)$ is the visibility prior. We propose to consider visibility as dependent on $\mathcal{D}$, to enable geometric reasoning on occlusions (section 2.4). In the E-step of the EM algorithm described below, this geometric visibility prior will be probabilistically mixed with photometric evidence, giving an estimate of the visibility that is more robust to geometric occlusions than using a uniform prior [13].
5. $p(\mathcal{I}|\mathcal{V},\mathcal{I}^*,\mathcal{D},\tau)$ is the likelihood of the input images. Particular attention is paid to this term (section 2.3), because we find that usual formulae are not satisfactory for the wide-baseline case.

The variables can be classified in three groups: the known variables (or data) $\mathcal{I}$ and τ, the wanted variables (or model) $\theta = (\mathcal{I}^*, \mathcal{D})$ and the hidden ones $\mathcal{V}$. The inference problem is now stated as finding the most probable value of the wanted variables, given the value of the known ones and marginalizing out the hidden ones. That is, we want to estimate

$$\theta^* = \arg\max_\theta p(\theta|\mathcal{I},\tau) = \arg\max_\theta \int p(\mathcal{I},\mathcal{V},\mathcal{I}^*,\mathcal{D},\tau)d\mathcal{V}$$

The following sections give a form to each term of the decomposition.

2.3. Likelihood

Pixels in input images are treated as noisy observations of the model. We suppose the noise to be independently identically distributed. The likelihood can be decomposed as the product of the per-pixel likelihoods:

$$p(\mathcal{I}|\mathcal{V},\theta) = \prod_i \prod_{\mathbf{x}} p(\mathcal{I}_i(\mathbf{x})|\mathcal{V},\theta) \qquad (2)$$

Note that this product is extended over the pixels in the input images and not over the points in the 3D model, as opposed to many of the previous works on Bayesian modeling of the stereo problem that define the likelihood as

$$p(\mathcal{I}|\mathcal{V},\theta) = \prod_{\mathbf{X}} \prod_i p(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}),\mathcal{V}) \qquad (3)$$

Although this has the great advantage of clearly representing the contribution of every model point to the total likelihood, it is, strictly speaking, incorrect.

The problems related to this approximation are sketched in figure 3. In the first case, many 3D points instantiated by the first image's depth map project to the same pixel in the second image. Computing the product over the 3D points as in (3) will overuse the second image's pixel. This is not a good idea given that the viewing angle of this pixel is really steep, hence its color is quite random and depends on the camera sensors. In the second case, only a few points of the first image's depth map project to the second image, so many pixels of the second image will be unused even if these pixels were seeing the scene better than any other.

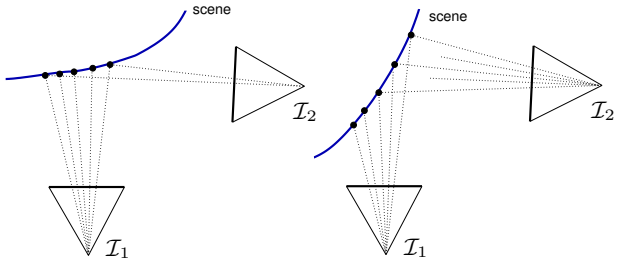

Figure 3. On the left, many 3D points instantiated by the first image project to the same pixel in the second one. On the right, many pixels on the second image have no 3D point instantiated by the first image that is projected onto them.

In small-baseline situations, where there is almost a bijection between pixels in each image and 3D model points from any other image's depth map, these effects are minimal and can be ignored. However, in our wide-baseline applications it is desirable to deal with them. In the following, we propose an approximation to the per-pixel product likelihood (2).

The per-pixel likelihood $p(\mathcal{I}_i(\mathbf{x})|\mathcal{V},\theta)$ measures the similarity between the color $\mathcal{I}_i(\mathbf{x})$ observed in the pixel $\mathbf{x}$ of image i, and the color that the model would predict for that pixel, let us call it $C_i^*(\mathbf{x})$. Remember that all 3D points are used to explain all images, hence $C_i^*(\mathbf{x})$ is computed from the colors of all 3D points that are projected onto that pixel, and may be different from $\mathcal{I}_i^*(\mathbf{x})$. Let us call $S_{i,\mathbf{x}}$ the set of points that are projected to $\mathbf{x}$ in image i. The color $C_i^*(\mathbf{x})$ is hard to define, because $S_{i,\mathbf{x}}$ may contain many points; its definition corresponds to a rendering problem. It seems natural to define $C_i^*(\mathbf{x})$ as the mean color of all visible ($\mathcal{V}_{i,\mathbf{X}} = 1$) points in $S_{i,\mathbf{x}}$: since they are currently considered to be visible by the pixel, they should contribute to predicting its color. Sadly, the resulting expression of the likelihood is difficult to deal with and in particular, the EM formulas become intractable.

To approximate this solution with a more usable expression, we define the per-pixel likelihood as the geometric mean of the likelihoods that the pixel would have if only one of the points in $S_{i,\mathbf{x}}$ was used,

$$p(\mathcal{I}_i(\mathbf{x})|\theta) = \prod_{\mathbf{X}\in S_{i,\mathbf{x}}} p(\mathcal{I}_i(\mathbf{x})|C(\mathbf{X}),\Sigma)^{\frac{1}{|S_{i,\mathbf{x}}|}}.$$

Computing the geometric mean of probabilities is equivalent to computing the arithmetic mean of energies. The idea behind is to cut the pixel's information in $|S_{i,\mathbf{x}}|$ parts and give one to each point in $S_{i,\mathbf{x}}$. This is justified as a manner

of using all the points in $S_{i,\mathbf{x}}$ without overusing the pixel $\mathbf{x}$. It is a heuristic approximation of the correct solution (2) but it solves the problems commented above and permits writing the likelihood as a per-point product

$$p(\mathcal{I}|\theta) = \prod_{\mathbf{X}} \prod_{i} p(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}), \Sigma)^{\frac{1}{|S_{i,\mathbf{x}}|}} \quad (4)$$

We refer to the term $p(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}), \Sigma)$ as the *pixel-point likelihood* and we model it by a mixture between a normal distribution in the case that $\mathcal{V}_{i,\mathbf{X}} = 1$ and a uniform distribution over the color space in case that $\mathcal{V}_{i,\mathbf{X}} = 0$. Since we work with probabilities for the visibility variables, this is:

$$\begin{aligned} p(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}), \Sigma) &= p(\mathcal{V}_{i,\mathbf{X}} = 1|\mathcal{D})\mathcal{N}(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}), \Sigma) \\ &+ p(\mathcal{V}_{i,\mathbf{X}} = 0|\mathcal{D})\mathcal{U}(\mathcal{I}_i(\mathbf{X})) \end{aligned} \quad (5)$$

When the prior on the visibility variables is constant, this distribution is called a *contaminated Gaussian* [16]. The following section describes the non-constant form that we give to this visibility prior.

2.4. Geometric Visibility Prior

The mixture of the pixel-point likelihood (5) is balanced by the visibility prior $p(\mathcal{V}_{i,\mathbf{X}}|\mathcal{D})$. This models the prior belief on whether the point $\mathbf{X}$ is visible or not in image $\mathcal{I}_i$, before taking into consideration the colors $C(\mathbf{X})$ or $\mathcal{I}_i(\mathbf{x})$. A uniform distribution is usually used for such a situation [13, 17]. However, our decomposition (1) of the joint probability, allows using the depth maps' information to give a more interesting form to this prior.

$\mathcal{D}_i(\mathbf{X})$ is the estimated depth of the pixel in image $\mathcal{I}_i$ onto which $\mathbf{X}$ is projected, which is not the same (see section 2.1) as the actual depth $d_i(\mathbf{X})$ of $\mathbf{X}$. If $d_i(\mathbf{X})$ is similar to $\mathcal{D}_i(\mathbf{X})$, it suggests that $\mathbf{X}$ is near the point seen by $\mathbf{x}$, so it should be more likely that $\mathbf{X}$ is visible. Symmetrically, if $d_i(\mathbf{X})$ is very different from $\mathcal{D}_i(\mathbf{X})$ the idea of image $\mathcal{I}_i$ seeing $\mathbf{X}$ seems unlikely. Thanks to this simple observation the geometric visibility can be easily and efficiently handled, in a multiple depth map approach. In [17] a threshold was used to strictly determine the visibility. Here we quantify the above idea by the (smooth) expression

$$p(\mathcal{V}_{i,\mathbf{X}} = 1|\mathcal{D}) = v \exp\left(-\frac{(d_i(\mathbf{X}) - \mathcal{D}_i(\mathbf{X}))^2}{2\sigma^2}\right)$$

where $v \in [0, 1]$ is the visibility prior for points at the estimated depth $\mathcal{D}_i(\mathbf{X})$ and σ models the tolerance that we give to points that are not exactly at this depth.

The effect of this prior on the pixel-point likelihood is in agreement with the above intuition. For points near the depth $\mathcal{D}_i(\mathbf{X})$, the prior is large and the normal distribution centered at $C(\mathbf{X})$ of the pixel-point likelihood mixture (5) is weighted up. This makes pixel colors similar to $C(\mathbf{X})$ more probable. For points far from the depth $\mathcal{D}_i(\mathbf{X})$, the uniform distribution is favored. The color $C(\mathbf{X})$ becomes irrelevant, which is logical given that we don't believe that the pixel $\mathrm{P}_i\bar{\mathbf{X}}$ is seeing the point $\mathbf{X}$.

2.5. Multiple Depth Map Prior

The multiple depth map prior $p(\mathcal{D}|\tau)$ is supposed to evaluate the plausibility of a set of depth maps without using any other information but the depth maps themselves. Two main properties are desired:

1. Each depth map should be mostly smooth but (strong) discontinuities have to be allowed.
2. The 3D points clouds belonging to the different depth maps should be *overlapping*.

Instead of using separate terms to measure smoothness and overlap, we evaluate the two properties in a single expression. To do so, we think of the set of depth maps as a point cloud forgetting, for a moment, the 2D neighborhood relation existing in the images. Smoothness and overlap will be reached by letting points attract one another, independently if they originate from the same depth map or not.

We express the probability of the point cloud as a Markov network:

$$p(\mathcal{D}) \propto \prod_{\mathbf{X} \in \mathcal{D}} \prod_{\mathbf{Y} \in N(\mathbf{X})} \varphi(\mathbf{X}, \mathbf{Y}) \quad (6)$$

where $N(\mathbf{X})$ denotes the neighborhood of $\mathbf{X}$ and $\varphi(\mathbf{X}, \mathbf{Y})$ is the compatibility probability for the $(\mathbf{X}, \mathbf{Y})$ pair. For the moment, this neighbourhood extends to the totality of points, $N(\mathbf{X}) = \mathcal{D} \setminus \{\mathbf{X}\}$.

Like for the pixel-point likelihood (5), we model the compatibility probabilities as mixtures of a normal and a uniform distribution, balanced by a hidden line process $\mathcal{L}$:

$$\begin{aligned} \varphi(\mathbf{X}, \mathbf{Y}) &\propto p(\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 1)\,\mathcal{N}(\mathbf{Y}|\mathbf{X}, \sigma') \\ &+ p(\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 0)\,\mathcal{U}(\mathbf{Y}) \end{aligned}$$

where $p(\mathcal{L}_{\mathbf{X},\mathbf{Y}})$ is the constant prior on the line process. $l = p(\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 1)$ is a parameter of the method. σ' is the variance of the isotropic three dimensional normal distribution $\mathcal{N}$. $\mathcal{U}$ is a uniform distribution over a volume containing the scene ($\mathcal{U}(\mathbf{Y}) = \mathcal{U}(\mathbf{X})$).

The underlying idea is that the process $\mathcal{L}_{\mathbf{X},\mathbf{Y}}$ signals if the two points should attract each other or not. If $\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 1$, we regard $\mathbf{Y}$ as a noisy measurement of $\mathbf{X}$ and its probability distribution is set to a normal distribution centered on $\mathbf{X}$ and with variance σ'. Note that this relationship is symmetrical. Otherwise, if $\mathcal{L}_{\mathbf{X},\mathbf{Y}} = 0$ a uniform distribution is used to reflect the idea that $\mathbf{X}$ and $\mathbf{Y}$ are not related.

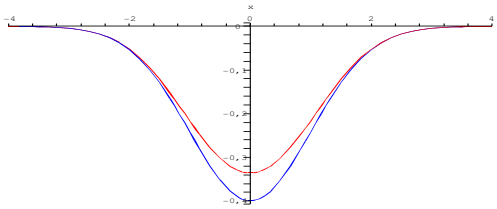

Figure 4. In red, a plot of the clique potentials of our prior, $-\log(\mathcal{N}(x|0,1)+1)$. In blue, the kernel correlation based one, $-\mathcal{N}(x|0,1)$.

This prior is computationally expensive. If m is the number of points, there are $O(m^2)$ compatibility probabilities. However, for all the points far enough from $\mathbf{X}$, $\mathcal{N}(\mathbf{Y}|\mathbf{X},\sigma')$ will be very small and $\varphi(\mathbf{X},\mathbf{Y})$ will be constant. We can thus restrict the neighborhood to the points near enough to $\mathbf{X}$. We define the neighborhood as the points inside a sphere centered at $\mathbf{X}$ with a radius ρ dependent on σ'. Finding this neighborhood is in itself a hard problem that can be expensive. Luckily, our point cloud comes from a set of depth maps where points are ordered. The projection of the neighborhood sphere in each image is an ellipse. The set of 3D points instantiated by the pixels inside these ellipses contain all neighbors of $\mathbf{X}$, greatly facilitating the task of finding them.

As desired, the proposed prior smoothes and integrates all the depth maps at the same time. Discontinuities are allowed thanks to the hidden line process $\mathcal{L}$ that avoids distant points to attract one another.

Kernel Correlation. Our prior is closely related to *leave-one-out* kernel correlation. Tsin and Kanade showed the capacities of the KC prior in smoothing while keeping discontinuities and applied it successfully to the stereo problem [18]. The KC prior can be written as a Markov network with

$$\varphi_{KC}(\mathbf{X},\mathbf{Y}) \propto exp(\mathcal{N}(\mathbf{X}|\mathbf{Y},\sigma'))$$

In figure 4, the negative logarithms of our compatibility probability and the KC-based one are plotted to show the similar shape they have. The advantage of the mixture prior over the KC is that it is defined in a probabilistic framework that permits the incorporation of new cues of information. We could, for example, use a statistical relation between the color of points and the line process $\mathcal{L}$, that makes points of the same color have a better chance to be attracted to one another.

2.6. Optimization

We maximize the posterior probability with the Expectation Maximization algorithm [2]. Direct non-linear optimization of our posterior is not only possible but also less expensive than EM. However, EM is known to often be more stable and easier to monitor as hidden variables are explicitly estimated. EM alternates between estimating the hidden variables' probabilities and optimizing the model. We start with a given initial model θ^0 (see section 3) and repeat the next steps until convergence.

E-step. In the expectation step we compute the posterior probabilities of our hidden variables $\mathcal{V}$ given the current estimate of the model. We store them as a set of visibility maps $f_{i,\mathbf{X}} = p(\mathcal{V}_{i,\mathbf{X}} = 1|\mathcal{I},\theta^t)$ and, by Bayes' rule,

$$f_{i,\mathbf{X}} = \frac{p(\mathcal{V}_{i,\mathbf{X}}=1|\mathcal{D})\mathcal{N}}{p(\mathcal{V}_{i,\mathbf{X}}=1|\mathcal{D})\mathcal{N} + p(\mathcal{V}_{i,\mathbf{X}}=0|\mathcal{D})\mathcal{U}}$$

where $\mathcal{N} = \mathcal{N}(\mathcal{I}_i(\mathbf{X})|C(\mathbf{X}),\Sigma)$ and $\mathcal{U} = \mathcal{U}(\mathcal{I}_i(\mathbf{X}))$ (see (5)). It is at this moment that the geometric visibility prior is mixed with the photometric evidence to give an estimation of the current visibility.

M-step. In the maximization step the expected visibility maps are used to maximize the expected log-posterior,

$$\theta^{t+1} = \arg\max_{\theta} \langle \log p(\mathcal{I}|\mathcal{V},\theta)\rangle_f + \log p(\mathcal{D})$$

i.e. the sum of the expected log-likelihood (cf. (4) and (5)),

$$\sum_{\mathbf{X}}\sum_{i} \frac{1}{S_{i,\mathbf{X}}} \left(f_{i,\mathbf{X}} \log\mathcal{N} + (1-f_{i,\mathbf{X}})\log\mathcal{U}\right)$$

and the log-prior (cf. (6)), $\sum_{\mathbf{X}}\sum_{\mathbf{Y}} \log\varphi(\mathbf{X},\mathbf{Y})$.

The maximum is searched by gradient descent. Analytical derivation of the log-posterior with respect to the model variables can be easily computed from the above equations. In our implementation, only one gradient descent iteration is done at each M-step. The iteration finds a better guess for θ^{t+1} but not the best. This method is called the Generalized EM algorithm. The motivation for doing this is that each iteration of the gradient descent method is as expensive as an E-step. Rapid alternation between E and M steps permits a faster actualization of the visibility maps.

3. Experiments

We have implemented the algorithm in a pyramidal scheme to speed up convergence and reduce the probability of being trapped in irrelevant local minima. We start using reduced versions of the original input images, and thus reduced versions of the colored depth maps. When convergence of EM is achieved, a higher resolution level is initialized with the obtained results, using bilinear interpolation.

In all our experiments, the noise variance Σ (see section 2.3) was included to the wanted variables and estimated during the optimization process. The visibility prior, v, was set to 0.9 expressing the idea that a point is likely to be visible in an image if it is at a similar depth to that estimated for that image (see section 2.4). σ' was set to the same value as

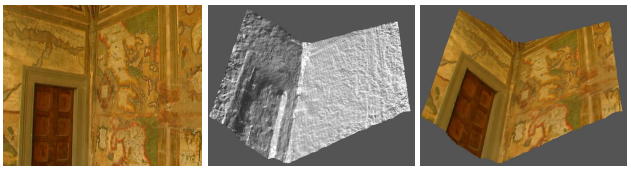

Figure 5. **Loggia**: One of the three input images (left) and renderings of its recovered depth maps.

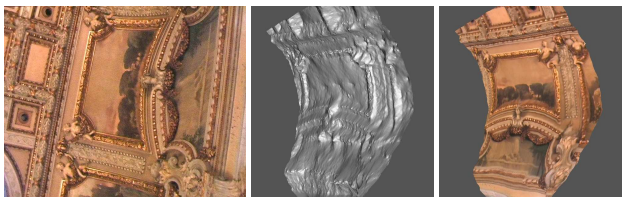

Figure 6. **Casino**: One of the five input images on the left and untextured and textured renderings of the recovered surface viewed from a very different angle.

σ (see sections 2.4 and 2.5). This value was heuristically set to two times the robust mean of the distance between pairs of 3D points instantiated from consecutive pixels in the images. The parameter l (see section 2.5) was the only one to be specially adapted for each experiment. We present the results on several datasets of increasing complexity.

Easy. The Loggia data set (figure 5) consists of three wide-baseline images of a scene with rich textures and simple geometry. Initial depth maps were set to a constant value (i.e. fronto-parallel) and the algorithm converged to the correct surface. The Casino data set (figure 6) contains five images with small baseline. Constant depth initialization was also used. The results show the potential of the method in capturing fine details. In both cases, large enough values of l ($l > 0.1$) gave similar results.

Medium. We tested our method's performance for the Cityhall scene [1] to prove that the algorithm can achieve state-of-the-art results in wide-baseline matching but with several depth maps at once. Images 3, 4 and 5 of the dataset were used. In this case, the model was initialized using the 3D feature point positions from the calibration step. Pixels with known depth were fixed while successive Gaussian blurs were applied to the rest of the depth map pixels. From this coarse initialization the algorithm converged, merging the depth maps into a single surface. The results (figure 7) show fine and rich details and the strong discontinuity between the foreground statues and the door was preserved.

Hard. To show the potential of the algorithm in dealing with strong discontinuities and geometric occlusions, we tested its performance on the challenging statue data set (figure 8). The scene contains a statue in front of a far wall. A single depth map is not enough to model the scene because none of the images sees the whole statue or wall. We used the same coarse initialization method as for the Cityhall scene.

[1] The Cityhall images with full calibration can be downloaded from `http://www.esat.kuleuven.ac.be/~cstrecha/testimages/`

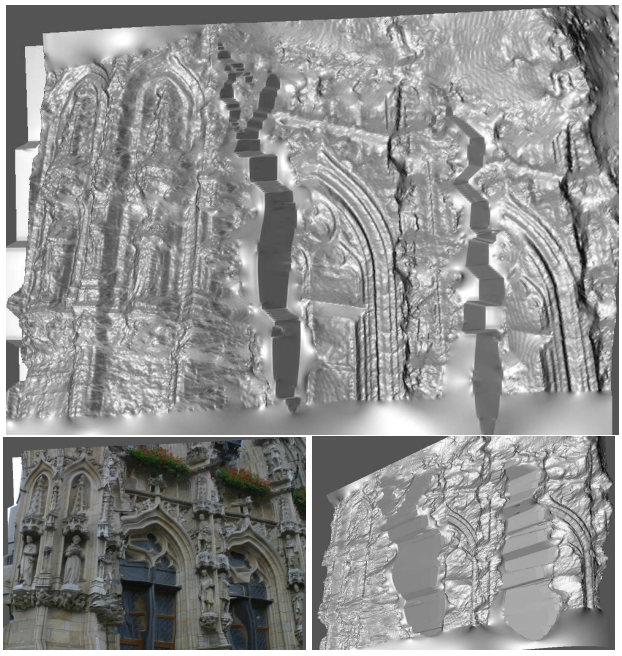

Figure 7. **Cityhall**: Untextured, textured and relighted renderings of an estimated depth map viewed from two different angles. No points were removed. The oversmoothed part at the bottom of the model corresponds to points seen only in one image. The two flat regions in the center correspond to discontinuities of the depth map.

The main difficulty was to strictly estimate the large discontinuity between the statue and the wall. Smoothing in this region would produce incorrect 3D points between the foreground and the background. We set the l parameter to a small value ($l = 0.2$) to motivate the points not to attract each other too much (see section 2.5). The discontinuity was then well preserved, but not at the exact position. Some background points remained attached to the statue. In addition, when initializing a finer level of the pyramid from a coarser one, we used bilinear interpolation which smoothed out the discontinuity.

To solve these problems we alternated several EM iterations with the following heuristic global search. For each pixel $\mathbf{x}$ and image i, we consider all the depths of the 3D points $S_{i,\mathbf{x}}$ that are projected to that pixel (see section 2.3). Then we test if the likelihood will be improved if we change the depth of pixel $\mathbf{x}$ to any of these values. The value producing the best improvement is kept. The large discontinuity between the statue and the wall was detected by the EM algorithm from the coarser level. The global search heuristic placed this discontinuity at the correct position and mantained it there in the finer levels.

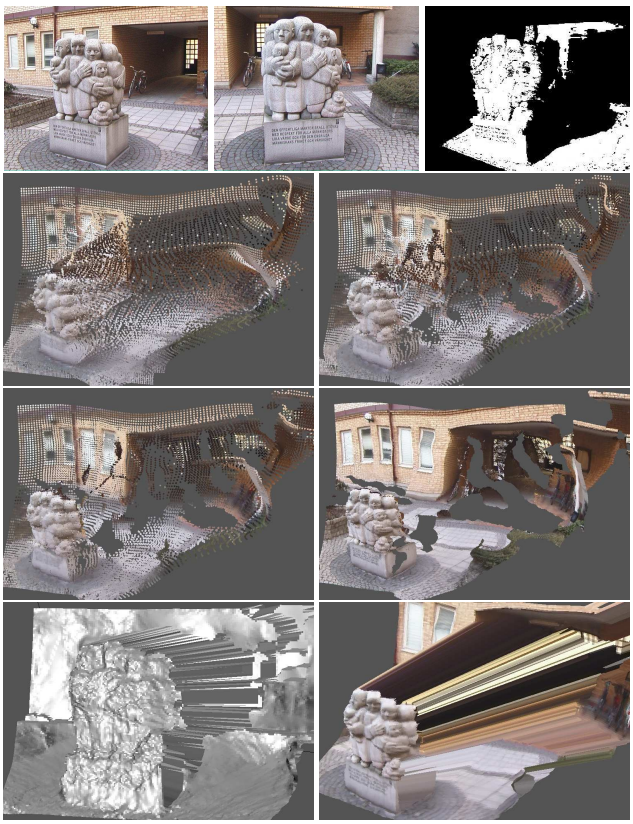

Figure 8. **Statue**: On top, first and last of the five input images and the visibility map of the first image with respect to last, i.e. the estimated probabilities of the 3D points instantiated by the first depth map to be visible in the last image. The next two rows show a point rendering of the set of all the depth maps at the same time during the evolution of the algorithm, from a very coarse initialization, to the final model. On the last row, two renderings of the estimated depth map $\mathcal{D}_2$ are shown. Note the well-preserved large discontinuities between statue and background.

4. Discussion

The proposed method was formulated in a rigorous probabilistic framework extending previous works. The experiments proved the pertinence of this extensions. However, there are still some issues to solve in order to make the method more usable.

The probabilistic approach permits the parameters of the method to be learned during the optimization. In effect, treating the parameters as random variables we can either estimate their most probable value or marginalize them out. Our current implementation needs to manually set three parameters. Although these parameters represent well defined concepts it will be preferable that the algorithm automatically sets them.

The other issue of the method, like in any other gradient descent based method, is the initialization. The pyramidal implementation of the EM algorithm converges well in cases where the strong discontinuities are captured from earlier small resolution levels. However, without a good initialization, it seems likely that for images such as the ones used in [12], the EM algorithm does not reach the global optimum but a local one. Interestingly, one of the best performing methods in this field [15], uses the same Bayesian scheme, but the optimization is done with the Loopy Belief Propagation algorithm. It is our interest to study the possibility of applying this or other global maximization techniques to our posterior probability definition.

Acknowledgements. This work used resources developed by partners of the European project VISIRE (IST-1999-10756). We especially would like to thank Martin Johansson and Anders Heyden for providing us with the data for the statue sequence.

References

[1] A. Broadhurst, T. W. Drummond, R. Cipolla. A probabilistic framework for space carving. *ICCV*, 2001.

[2] A. P. Dempster, N. M. Laird, D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.

[3] O. Faugeras, R. Keriven. Complete dense stereovision using level set methods. *ECCV*, 1998.

[4] A. Fitzgibbon, Y. Wexler, A. Zisserman. Image-based rendering using image-based priors. *ICCV*, 2003.

[5] W. T. Freeman, E. C. Pasztor. Learning low-level vision. *IJCV*, 40:25 – 47, 2000.

[6] P. Fua, Y. Leclerc. Object-centered surface reconstruction: combining multi-image stereo shading. *IUW*, 1993.

[7] V. Kolmogorov, R. Zabih, S. J. Gortler. Generalized multi-camera scene reconstruction using graph cuts. *EMMCVPR*, 2003.

[8] K. Kutulakos, S. Seitz. A theory of shape by space carving. *IJCV*, 38(3):199–218, 2000.

[9] D. Morris, T. Kanade. Image-consistent surface triangulation. *CVPR*, 2000.

[10] S. Paris, F. Sillion, L. Quan. A surface reconstruction method using global graph cut optimization. *ACCV*, 2004.

[11] M. Pollefeys. *Self-Calibration and Metric 3D Reconstruction from Uncalibrated Image Sequences*. PhD thesis, 1999.

[12] D. Scharstein, R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47:7–42, 2002.

[13] C. Strecha, R. Fransens, L. Van Gool. Wide-baseline stereo from multiple views: a probabilistic account. *CVPR*, 2004.

[14] C. Strecha, T. Tuytelaars, L. Van Gool. Dense matching of multiple wide-baseline views. *ICCV*, 2003.

[15] J. Sun, H.Y. Shum, N.N. Zheng. Stereo matching using belief propagation. *PAMI*, 25(7), July 2003.

[16] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *IJCV*, 5(3):271–301, 1990.

[17] R. Szeliski. A multi-view approach to motion and stereo. *CVPR*, 1999.

[18] Y. Tsin, T. Kanade. A correlation-based model prior for stereo. *CVPR*, 2004.

[19] G. Vogiatzis, P.H.S. Torr, R. Cipolla. Bayesian stochastic mesh optimization for 3d reconstruction. *BMVC*, 2003.