Robust One-Shot 3D Scanning Using Loopy Belief Propagation

Ali Osman Ulusoy* Fatih Calakli* Gabriel Taubin

Brown University, Division of Engineering
Providence, RI 02912, USA
{ali_ulusoy, fatih_calakli, taubin}@brown.edu

Abstract

A structured-light technique can greatly simplify the problem of shape recovery from images. There are currently two main research challenges in design of such techniques. One is handling complicated scenes involving texture, occlusions, shadows, sharp discontinuities, and in some cases even dynamic change; and the other is speeding up the acquisition process by requiring small number of images and computationally less demanding algorithms. This paper presents a "one-shot" variant of such techniques to tackle the aforementioned challenges. It works by projecting a static grid pattern onto the scene and identifying the correspondence between grid stripes and the camera image. The correspondence problem is formulated using a novel graphical model and solved efficiently using loopy belief propagation. Unlike prior approaches, the proposed approach uses non-deterministic geometric constraints, thereby can handle spurious connections of stripe images. The effectiveness of the proposed approach is verified on a variety of complicated real scenes.

1. Introduction

Three-dimensional (3D) shape acquisition from images has been one of the most important topics in computer vision due to its wide range of applications in engineering, entertainment, visual inspection and medicine. It has especially been instrumental in digitizing and preserving archaeological artifacts. Although numerous 3D reconstruction methods have been proposed for this purpose, there isn't *one* solution that is suitable for all sorts of archaeological objects. Challenges include shadowy areas, immobility of the object, complex geometry that prevents reaching certain areas, fragile nature of old artifacts, etc. Some of these challenges can be alleviated by a hand-held scanner which contains one or more cameras and possibly a laser emitter or a projector. It is capable of scanning objects when manually swept over them.

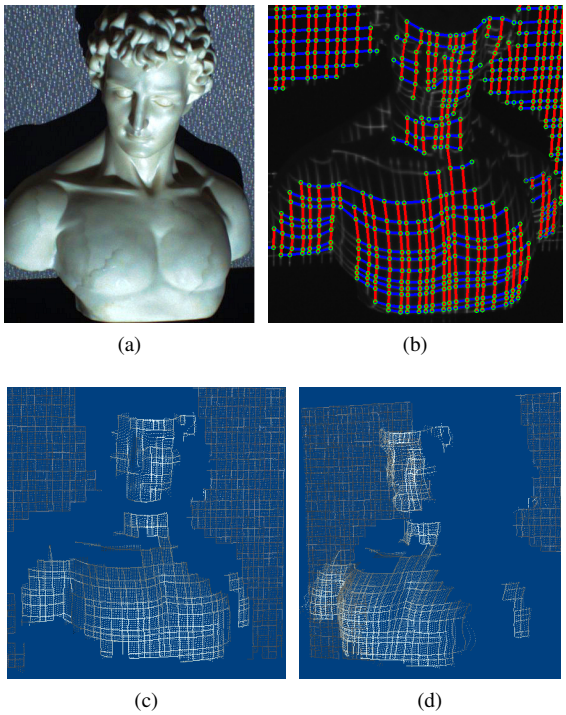


Figure 1: (a) A bust, (b) the intersection network overlaid onto the image, (c,d) reconstructions from two views.

Typically, a hand-held scanner works by continuously registering a sequence of 3D views as they are acquired. Obviously, the 3D acquisition has to be done very quickly (ideally within a single frame) as the scanner is constantly moving. Most commercial hand-held scanners use a single laser beam for this purpose which can reconstruct only a single slit per frame. In such systems, the scanner has to be swept very slowly to capture all the intricate details. Also, since the reconstruction is sparse, the registration step can easily fail if the scanner moves too fast or makes an abrupt motion.

In this paper, we propose using a grid pattern to handle some of these challenges. This pattern allows for build-

*First two authors contributed equally to this work and order of authorship was decided through flipping a coin.

ing a "one-shot" system while still maintaining a dense reconstruction. We argue that previous approaches for grid based systems are not adequate for dealing with scenes of complex geometry. To alleviate this, we introduce a novel probabilistic graphical model and show that it can deal with such scenes. Combined with an online registration method, we think that this method shows great potential for use in a hand-held scanner.

2. Related Work

There exists several 3D reconstruction techniques such as structured light, multi-view stereo vision and photometric stereo. Structured light techniques have been very popular in applications to archeology due to their accuracy, density and ease of use [9, 3]. Since a high variety of approaches have been proposed to this day, we will only discuss some of the most relevant ones here. A comprehensive assessment is presented in [14, 1] by Salvi *et al*.

Most of the early work in structured lighting follow a temporal approach: multiple patterns are projected consecutively onto the object and a camera captures an image for each projected pattern. These methods require the object to be perfectly still while the patterns are being projected. Obviously, this is not acceptable for a hand-held device as the scanner is constantly moving. If however, the number of patterns to be projected is reduced to a single pattern, then such methods become feasible. Although there are also techniques that use a single but adaptive pattern [7], they require a video projector. In this study, we focus and review methods that project a single *static* pattern in which case a slide or laser projector would suffice.

Numerous one-shot patterns have been proposed so far. One of the simplest approaches is to project a single slit, typically using a laser beam. Although simple and easy to implement, these approaches suffer from sparsity, *i.e*., only a single slit is reconstructed at each frame. A natural extension is to project multiple slits or to have both horizontal and vertical lines which form a grid pattern. In [12], Ru and Stockman suggest using such a grid pattern. As for their solution, the observed grid in the camera is matched to the projected pattern by exploiting a number of geometric and topological constraints. A major drawback is that the detected grid points are numbered with respect to a reference point, *i.e*. relative numbering. This necessitates that at least a patch of the pattern be extracted perfectly since in the case of undetected or spurious grid points, the algorithm will fail. Although Proesmans *et al*. [11] present a more efficient solution to the grid based systems, they suffer from relative numbering also.

Another popular approach to obtain a dense reconstruction is the use of colors [18, 4]. These approaches typically use complex illumination patterns and are highly sensitive to noise and object texture. Therefore they require sophisticated image processing and possibly color calibration [4].

Recently, Kawasaki *et al*. [5] proposed a two color grid where vertical and horizontal lines are of different colors. Unlike most other previous approaches, their method does not rely heavily on image processing and is robust against undetected grid points as it does not assume relative ordering. Their solution includes singular value decomposition (SVD) on a large and very sparse matrix, which is numerically instable. In [16], Ulusoy *et al*. propose a more stable solution with the help of a special grid. Both methods heavily exploit the coplanarity constraints that arise due to the links between two grid points. However, such constraints are often violated because spurious links between grid points are observed in complex scenes. It is very important to note that even a *single* spurious link typically causes the entire 3D solution to be incorrect, rendering both these methods unsuitable for such scenes.

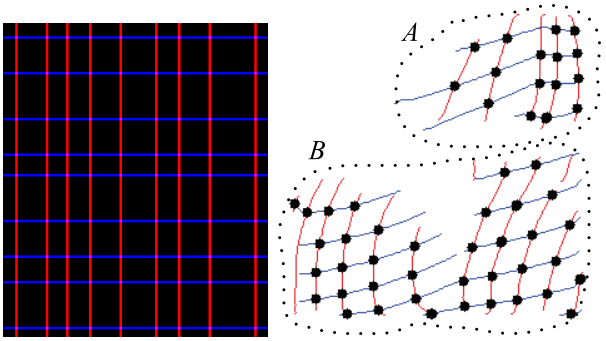
The primary contributions of this paper are: (1) formulation of the problem using a novel probabilistic graphical model, (2) handling spurious connections resulting from complicated scenes, (3) an efficient solution using loopy belief propagation.

3. Overview of the Approach

In this paper, we present a structured-light approach consisting of a projector, and a single color camera. Both are calibrated with respect to a world coordinate system. The projector projects a known grid pattern onto a 3D object. The camera captures an image of the object illuminated by the projected grid. This single image is used to determine the depth information of points illuminated by grid stripes observed by the camera. Producing the correct result depends on being able to solve the so-called *correspondence* problem, *i.e*., matching pairs of grid points and their images need to be identified.

At first, finding correspondences might seem to require an exhaustive search throughout the whole grid. However, geometric constraints greatly simplifies the problem. First, the projector-camera system adds *epipolar* constraints that reduce the search space of each grid point to a single line, *i.e*., the correspondence of the point can only come from one of grid crossings along its epipolar line. This is explained in Section 4.1.1. Second, the grid pattern discloses spatial neighborhood information for all captured grid points. This adds *coplanarity* constraints and *topological* constraints on their correspondences, *i.e*., if two grid points are linked horizontally on the captured image, then their correspondences must abide to certain constraints. These are explained in detail in Sections 4.1.2 and 4.1.3.

The problem is still challenging as difficulties arise, *i.e*., establishing correspondences can be easily tricked by a complex scene that involves texture, occlusion, shadows, sharp discontinuities, and dynamic change [7]. As a result,



(a) (b)

Figure 2: (a) A grid pattern, (b) two separate intersection networks.

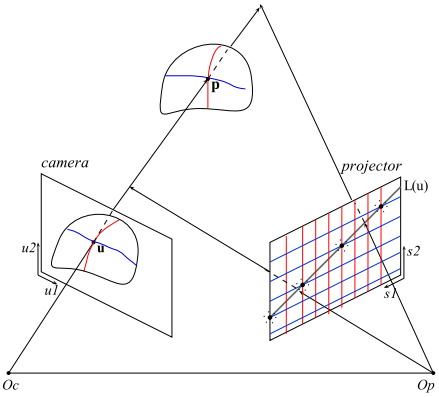


Figure 3: A point $\mathbf{u}$ and its epipolar line $L(\mathbf{u})$.

observed geometric constraints can be misleading which is explained in Section 4.1.4. We present a novel probabilistic graphical model in Section 4.2 that aims to formulate these geometric constraints in a stochastic fashion. After building the graphical model, the solution is obtained efficiently using loopy belief propagation.

4. Reconstruction using Grid Pattern

We assume that the camera and the projector are calibrated. The projector projects a grid pattern, composed of horizontal blue stripes and vertical red stripes as shown in Figure 2a, to the measured object and the camera is placed to detect it. A simple image segmentation process in the captured image, *e.g.*, thresholding color channels, differentiates between the horizontal and vertical stripes. The stripes are then combined to form an input to the system - *a 2D intersection network*. However, they, in general, may not result in a single intersections network due to shadows, occlusions and sharp depth discontinuities. An example depicting this is given in Figure 2b. Our approach solves the correspondence problem for each intersection network independently. This section explains the steps for solving the correspondence of a single intersection network. Without loss of generality, the same steps can be applied to all intersection networks independently.

As for the notation used in this section, all image measurements refer to the homogeneous normalized coordinates because the calibration is known for both camera and projector.

4.1. Labeling Problem

The grid pattern is a set of horizontal and vertical stripes. Suppose those stripes have no width, then we can refer to a stripe as a line π. The intersection of two perpendicular lines is a grid crossing $\mathbf{s} = (s_1, s_2, 1)^t$ in the projector image. Due to the projection, every grid crossing forms a light line and every grid line forms a light plane in 3-space. These are reflected on the object surface and identified in the camera image. The detection of such a light line is referred to as an *intersection* point $\mathbf{u} = (u_1, u_2, 1)^t$ in the camera image. The reflection of the light plane is referred to as a set of *links* that connect the neighboring intersection points. We treat the union of links and intersection points as an intersection network.

A grid crossing $\mathbf{s}$ parameterizes both the horizontal line π^h and the vertical line π^v it lies on. Therefore, a horizontal and a vertical triangulation plane is defined by the the grid crossing $\mathbf{s}$. Given an intersection $\mathbf{u}$, and its correspondence $\mathbf{s}$, all the points in the intersection network that are horizontally and vertically linked can be triangulated using optical ray-plane triangulation. Thus, in order to triangulate all the points in an intersection network, we only need correspondences $\mathbf{s}$ of intersection points $\mathbf{u}$. Formally, we are looking for a labeling function

$$f \ : \ \mathbf{u_i} \rightarrow \mathcal{S}, \ i = 1 \dots N \tag{1}$$

where $\mathcal{S} = \{\mathbf{s_1}, \mathbf{s_2}, \dots, \mathbf{s_M}\}$. N and M are the number of all intersection points and grid crossings respectively.

4.1.1 Epipolar Constraint

Assuming that the world coordinate system is placed at the optical center of the camera and ignoring intrinsic parameters, the equation of the projection of a 3D point $\mathbf{p} = (p_1, p_2, p_3)^t$ onto the normalized camera image point $\mathbf{u}$ is $\lambda\mathbf{u} = \mathbf{p}$. We define another coordinate system at the optical center of the projector for convenience and denote projector image point as $\mathbf{s}$. The relation between $\mathbf{u}$ and $\mathbf{s}$ is

$$\mu\mathbf{s} = \lambda R\mathbf{u} + T \tag{2}$$

where both λ and μ are unknown scalar values, R is the rotation matrix and T is the translation vector which define the coordinate transformation between the two coordinate systems.

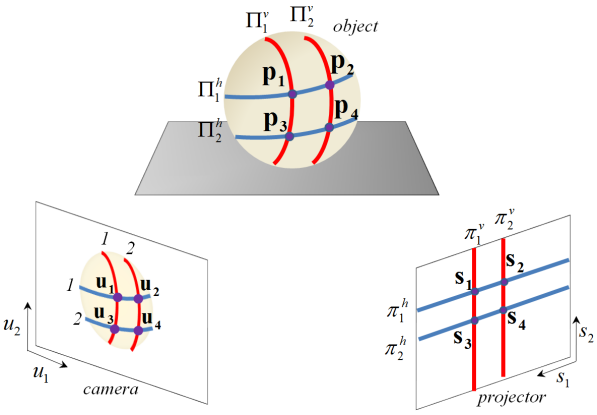


Figure 4: A typical scenario revealing coplanarity constraints and topological constraints

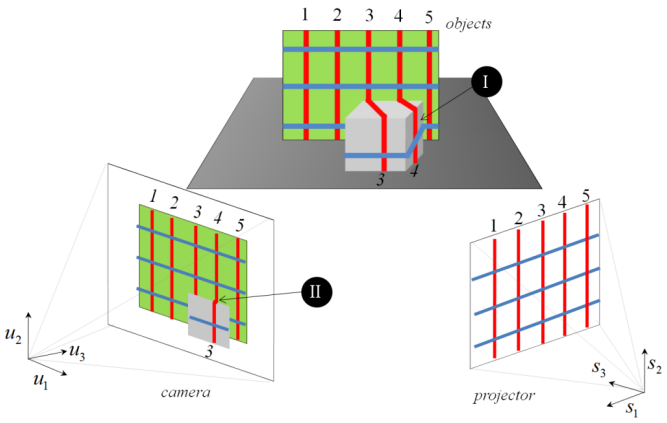


Figure 5: Challenges in finding correspondences, (I) an occluded stripe, (II) a spurious link.

By eliminating the unknown scalars λ and μ from equation (2), we retrieve the epipolar line L of the camera point $\mathbf{u}$ in projector image as

$$L(\mathbf{u}) = \{\mathbf{s} : l^t \mathbf{s} = 0\} \tag{3}$$

where $l = \hat{T}R\mathbf{u}$, and $\hat{T}$ is the matrix representation of the cross product with T. This representation follows that the Euclidean distance from a grid crossing $\mathbf{s}$ to the epipolar line $L(\mathbf{u})$ is

$$dist(s, L(\mathbf{u})) = |s^t L(\mathbf{u})|. \tag{4}$$

Equation (3) says that a captured grid crossing $\mathbf{u}$ in the camera image may correspond only to grid crossings on the epipolar line $L(\mathbf{u})$ in the projector image as shown in Figure 3. However, a correspondence typically does not lie exactly on the epipolar line due to imperfections in calibration procedure. Therefore, we define a threshold τ on the distance $dist(s, L(\mathbf{u}))$, and consider only the grid crossings $\mathbf{s} \in \mathcal{S}'$ whose distance is smaller than τ. Note that the size $\mathcal{S}'$ is dramatically smaller than that of $\mathcal{S}$.

4.1.2 Coplanarity Constraint

A grid pattern consists of a set of discrete horizontal lines $\pi^h_j,\ j = 1, 2, \ldots, H$, and a set of discrete vertical lines $\pi^v_k,\ k = 1, 2, \ldots, V$. The projection of a horizontal line π^h forms a horizontal plane Π^h in 3-space. Similarly, a vertical line π^v forms a vertical plane Π^v. The reflection of those planes are observed by the camera. In the example of Figure 4, an intersection network of four camera points $\mathbf{u_1}$ to $\mathbf{u_4}$ are given. These points correspond to 3D-points $\mathbf{p_1}$ to $\mathbf{p_4}$, and also to grid crossings $\mathbf{s_1}$ to $\mathbf{s_4}$. Consider the horizontal link between $\mathbf{u_1}$ and $\mathbf{u_2}$. Notice that the 3D points $\mathbf{p_1}$ and $\mathbf{p_2}$ which generated $\mathbf{u_1}$ and $\mathbf{u_2}$ lie on the same plane Π^h_1. This plane is originated by the horizontal line π^h_1 in the projector image. Therefore, the correspondences of $\mathbf{u_1}$ and $\mathbf{u_2}$ must lie on the horizontal line π^h_1. More generally, if two detected intersections are linked in the camera image, then their correspondences must be *collinear* in the projector image. If the link is horizontal, then their correspondences must have the same j value, where the integer j is the horizontal *line identifier* within the set of H horizontal grid lines. Similar constraints can be derived for the vertical case.

4.1.3 Topological Constraint

Again, consider the detected intersections $\mathbf{u_1}$ and $\mathbf{u_2}$ in the example of Figure 4. These points are linked horizontally in the camera image, and we know that their correspondences must be collinear. There is more than that. In the physical layout of the camera image points, $\mathbf{u_1}$ is on the left of $\mathbf{u_2}$. This is due to the fact that the 3D point $\mathbf{p_1}$ is also on the left of $\mathbf{p_2}$. Therefore, the correspondence of $\mathbf{u_1}$ must also be on the left of the correspondence of $\mathbf{u_2}$. More generally, if two intersections are linked in the camera image, then their *topological structure* is preserved for their correspondences in the projector image. If the link is horizontal, then the correspondence for the intersection on the left must have a lower k value than that of the intersection on the right, where the integer k is the vertical line identifier within the set of V vertical grid lines. Similar constraints can be derived for the vertical case.

4.1.4 Spurious Connection

Difficulties in finding the correct correspondences may arise, even when exploiting the geometric and topological constraints. Typical complications are shown in Figure 5, *i.e.*, a depth discontinuity that is parallel to the grid lines of the pattern may cause them to disappear in the camera image, whereas a depth discontinuity that is not parallel to the grid lines can let two different grid lines appear as a single link in the camera image, namely a *spurious link*. There-

fore, observed links in the camera image may be misleading for establishing correspondences.

A probabilistic graphical model is ideal for combining the aforementioned geometric and topological constraints, and also account for the non-deterministic nature of the problem.

4.2. Probabilistic Graphical Model Formulation

We start by interpreting the intersection network in the camera image as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each intersection is a node $v \in \mathcal{V}$ and links between intersections are edges $e \in \mathcal{E}$. We distinguish between the set of horizontal and vertical edges as $\mathcal{E}_{hor}$ and $\mathcal{E}_{ver}$ where $\mathcal{E} = \mathcal{E}_{hor} \cup \mathcal{E}_{ver}$. Next, we define $\mathcal{X} = \{X_1, X_2, ..., X_N\}$ to be N discrete valued random variables, where each random variable is associated with a node in $\mathcal{V}$. Random variable X_i represents the correspondence of an intersection $\mathbf{u_i}$ to a grid intersection $\mathbf{s}$ in the projector image. The labeling problem is to assign a state (correspondence) to each random variable.

In this section, we show that we can exploit the geometric constraints of our system easily by modeling them with a probabilistic graphical model. Furthermore, we realize that solving the correspondence problem becomes equivalent to finding the maximum a posteriori (MAP) configuration, for which we propose a solution based on loopy belief propagation [10].

We use the factor graph notation presented in [8] where each random variable is depicted as a *variable node* and functions of these variables are depicted as *factor nodes*, illustrated as black squares. Edges connect variables to factor nodes if and only if that variable is an argument of the function. We use the factor node and the function it represents interchangeably.

For our graph, we have two types of factor nodes. Unary factors connect to one variable node as they represent a function of a single random variable. This function models the epipolar line constraint and is denoted as $\psi_{epipole}$. Pairwise factors connect to two variable nodes as they are functions of a pair of random variables. These functions model both the coplanarity constraint and the topological constraint due to vertical and horizontal links and are denoted as ψ_{ver} and ψ_{hor} respectively. The joint probability can be written as a product of the unary and pairwise factors,

$$p(\mathcal{X}) \propto \prod_{i=1}^{N} \psi_{epipole}(X_i) \prod_{(i,j)\in\mathcal{E}_{hor}} \psi_{hor}(X_i, X_j) \prod_{(i,j)\in\mathcal{E}_{ver}} \psi_{ver}(X_i, X_j) \tag{5}$$

Note that we have introduced factor nodes ψ_{ver} and ψ_{hor} for each $e \in \mathcal{E}_{ver}$ and $e \in \mathcal{E}_{hor}$ respectively. An example factor graph construction is given in Figure 6.

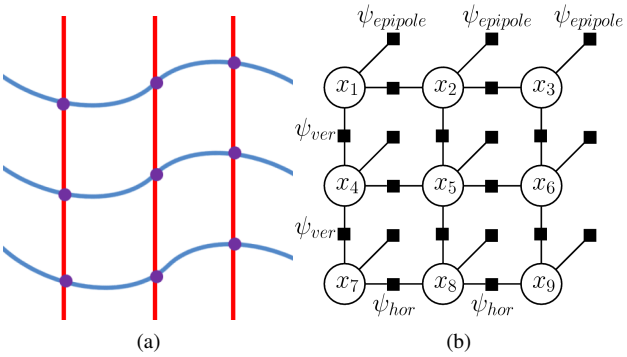


Figure 6: (a) An example intersection network, (b) its corresponding factor graph. Note that some of the factor labels were omitted for clarity.

The unary factors $\psi_{epipole}(X_i = \mathbf{s})$ models the epipolar constraint as described in Section 4.1.1. Note that associated with each random variable X_i we have a position $\mathbf{u_i}$ in the camera image. Correspondences sufficiently distant from the epipolar line get probability 0, *i.e.*, they are discarded entirely and the ones closer get a score based on their distance to the line. More formally, we have:

$$\psi_{epipole}(X_i = \mathbf{s}) = \begin{cases} 1/dist(\mathbf{s}, L(\mathbf{u_i})), & \text{if } dist(\mathbf{s}, L(\mathbf{u_i})) \leq \tau \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

The pairwise factors ψ_{hor} and ψ_{ver} model both the coplanarity constraint and the topological constraints due to the horizontal and vertical links in the camera image. The coplanarity constraint as described in Section 4.1.2 states that if two intersections are linked with a horizontal edge, then the respective two random variables must be assigned correspondences with the same horizontal line identifier, otherwise the compatibility is 0. If this constraint holds, then we check for the topological constraint which is described in Section 4.1.3 using the function ϕ. ψ_{hor} is formally defined in Equation 7. ψ_{ver} can be defined similarly.

$$\psi_{hor}(X_i = \mathbf{s}, X_j = \mathbf{s}') = \begin{cases} \phi\Big(sgn(\mathbf{u}_{i1} - \mathbf{u}_{j1}) \,.\, (id^v(\mathbf{s}) - id^v(\mathbf{s}'))\Big), & \text{if } id^h(\mathbf{s}) = id^h(\mathbf{s}') \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $id^h(\mathbf{s})$ and $id^v(\mathbf{s})$ are functions that take a grid intersection $\mathbf{s}$ and return the associated horizontal and vertical identifiers (described in Section 4.1.2) respectively. Note that if $\mathbf{u}_i$ is to the right of $\mathbf{u}_j$, the sgn function returns 1 and the argument $(id^v(\mathbf{s}) - id^v(\mathbf{s}'))$ is passed to ϕ (see Equation 8). As seen in Figure 7, the ϕ is nonzero only when there is a positive difference between the vertical line identifiers, *i.e.*, $\mathbf{s}$ is to the right of $\mathbf{s}'$. If $\mathbf{u}_i$ is to the left of $\mathbf{u}_j$, the sgn function corrects the argument to ϕ by reversing it.

$$\phi(\alpha) = \begin{cases} exp(1-\alpha), & \text{if } \alpha \geq 1 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

As seen in Figure 7, ϕ achieves its maximum when $\alpha = 1$. Notice that if ϕ were 0 for $\alpha > 1$, then our model wouldn't be able to handle occlusions, i.e. undetected grid points. We use an exponentially decaying function after $\alpha = 1$. This suggests that we do allow for large differences in vertical line identifiers but we penalize so as to keep the correspondences closer to each other. Note that a more forgiving function (less sharp than the exponential) could also be used.

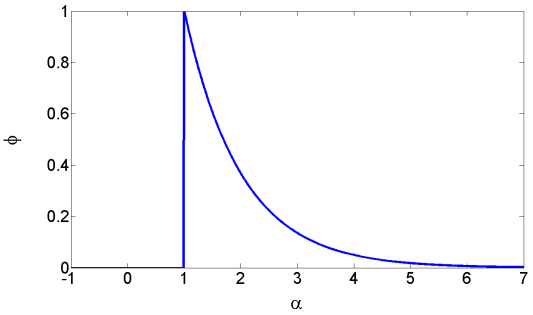


Figure 7: The ϕ function.

4.2.1 Dealing with spurious connections

To deal with these spurious connections described in Section 4.1.4, we introduce binary nodes b_k, $k = 1, ..., |\mathcal{E}|$ for each edge which are connected to the factor nodes as depicted in Figure 8a. We also define a prior λ_k for each b_k. When the $b_k = 1$, we assume the edge between X_i and X_j actually exists so we use the old compatibility measure ψ_{ver} or ψ_{hor}. However when $b_k = 0$, we assume the edge does not exist, in which case X_i and X_j become independent as there is no function binding them. So, we assign uniform compatibility to all possible pairs of values. This is formalized in Equation 9.

$$\psi'_{hor}(X_i, X_j, b_k) = \begin{cases} \psi_{hor}(X_i, X_j), & \text{if } b_k = 1 \\ c, & \text{otherwise} \end{cases} \quad (9)$$

where c is a constant. Note that for our purposes, we do not need to estimate b_k. Thus, we can simply sum them out as shown in Figure 8b. We end up with factor $\bar{\psi}_{hor}$ defined in Equation 10. One can derive $\bar{\psi}_{ver}$ similarly.

$$\bar{\psi}_{hor}(X_i, X_j) = \lambda_k \psi_{hor}(X_i, X_j) + (1 - \lambda_k)c \quad (10)$$

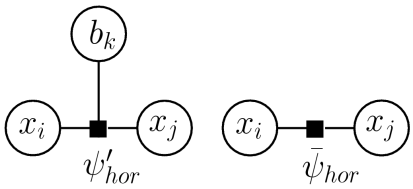


Figure 8: We sum over the binary edges as we do not need to estimate them. Unary factors have been omitted for clarity.

4.2.2 Loopy Belief Propagation: Max-Product

The correspondence problem for our model is finding the most likely assignment to each random variable, $\hat{\mathcal{X}} = \{\hat{X}_1, \hat{X}_2, ..., \hat{X}_N\} = \arg\max_{\mathcal{X}} p(\mathcal{X})$, *i.e.*, the MAP estimation problem. If our graph did not contain cycles (loops), we could have used the max-product algorithm [17] to get an exact answer. This algorithm is a variant of the sum-product algorithm [8] and belief propagation [10]. Unfortunately, our graph contains loops in which case the MAP estimation problem becomes NP hard and the max-product is no longer exact. However, approximate solutions can still be obtained by running it in an iterative fashion. Remarkably, max-product has been shown to be very effective in practice and is widely used [15]. There exists also theoretical work on explaining the empirical success of max-product [17].

Max-product is an iterative algorithm that works by passing messages in a graphical model [17]. Each message encodes the likelihood of a random variable using all the information that has propagated throughout the graph. Although both the sum-product and max-product algorithms can be defined for factor graphs [8], the equations can be simplified when the graph has only pairwise factors, *i.e.*, functions of up to two random variables. This is indeed the case for our graph so we will present the algorithm in this form.

A message at iteration t is defined as follows:

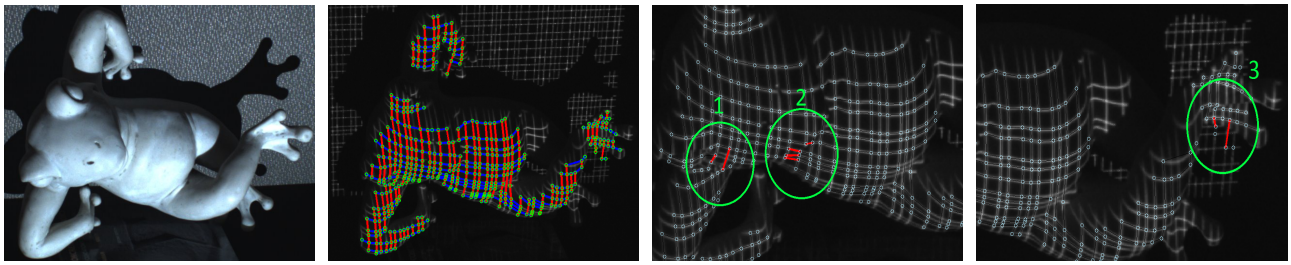
$$m^t_{i\to j}(X_i) \propto \quad (11)$$
$$\max_{X_i} \psi_{epipole}(X_i)\psi_{comp}(X_i, X_j) \prod_{k \in N(i)\setminus j} m^{t-1}_{k\to i}(X_i)$$

where ψ_{comp} denotes either horizontal ($\bar{\psi}_{hor}$) or vertical ($\bar{\psi}_{ver}$) compatibility depending on how X_i and X_j are connected. $N(i)\setminus j$ means the neighbors of i except for j. After passing messages around for t iterations, the MAP configuration can be computed using the equation below.

$$\hat{X}_i = \arg\max_{X_i} \psi_{epipole}(X_i) \prod_{k \in N(i)} m^{t-1}_{k\to i}(X_i) \quad (12)$$

5. De Bruijn Spaced Grids

It is known that when solving for correspondences using uniformly spaced grid patterns, one usually cannot



(a) (b) (c) (d)

Figure 9: (a) A comfortable frog, (b) the intersection network overlaid onto the image, (c,d) spurious connections.

obtain a confident solution and is left with ambiguities [5, 16, 13]. These patterns are especially ill-suited for complicated scenes with shadows, occlusions and sharp depth discontinuities. In such scenes, the grid pattern usually gets broken into multiple connected components as seen in Figure 2b. Most components contain a small number of intersections and links in which case there is very little geometric information to solve for correspondences. This information is usually not enough to obtain the correct solution.

A similar argument is made in [5] and irregular spacings are suggested to disturb the uniformity. The authors use a pattern with uniformly spaced vertical lines and randomly spaced horizontal lines. This indeed increases the robustness of the search, however, does not guarantee correct convergence for their case. Recently, De Bruijn spaced grids were proposed in [16]. These are grid patterns with spacings that follow a De Bruijn sequence. A k-ary De Bruijn sequence of order n is a cyclic sequence containing letters from an alphabet of size k, where each subsequence of length n appears exactly once.

We generate grid patterns where both the vertical and horizontal spacings between the stripes follow a De Bruijn sequence. Thus, a 2D patch consisting of n vertical spacings and n horizontal spacings and containing $(n+1)^2$ grid crossings is unique in the whole grid pattern. This makes sure that there is a unique configuration for which the joint probability is maximum, *i.e.*, the correct MAP estimate is unique.

6. Experiments and Results

We have implemented the proposed system using a 1024 x 768 DLP projector and a 1600 x 1200 CCD camera. Note that since we project a static pattern, we could have used a slide projector as well. We calibrated both the camera and the projector using the *Camera Calibration Toolbox for Matlab* [2]. The experiments were carried out under weak ambient light.

For the graphical model parameters, we have chosen binary edge prior $\lambda_k = 0.95 \ \forall k$ since we expect spurious connections to happen rarely. As for the constant c in Equation 9, we have realized that a small nonzero value such as $c = 0.01$ will suffice.

For the De Bruijn sequence parameters, we saw that $k = 5$ and $n = 3$ gave good results empirically as almost all detected patches were bigger than the needed size for unique identification and since k is not very large, we did not sacrifice much on reconstruction density. An example pattern is given in Figure 2a.

To assess the correctness of our results, we used a well established temporal structured lighting method [6] as ground truth. We ran both our method and the temporal method on static scenes and checked if all the intersections we found were assigned their true correspondences.

First, we present results for a complicated object with sharp depth discontinuities. The object and the intersection networks we capture are given in Figures 9a and 9b. The network (including the intersections on the background) is composed of 6 connected components. Four of these components contain less than 40 points. Figures 9c and 9d depict the spurious connections outlined in bold red. The first one is due to the sharp depth discontinuity between the head and the arm. Similarly, the second one is caused by the discontinuity between the head and the belly. The third one is more interesting: the foot gets linked to the background.

We have triangulated all the intersections and curves and then used interpolation to fill in between curve segments for visualization purposes. Our reconstruction results are given in Figure 10. Notice that the method has been able to handle the first two spurious connections: the arm, head and belly are separated even though they were connected to begin with. However, it had problems with the last one. This is expected since there is not enough geometry around the spurious link to force it to its correct position. In fact, there are only 2 intersections on one side of connection as seen in Figure 9d. Comparing with the ground truth, we saw that only these 2 intersections (out of 1015) were assigned wrongly. This corresponds to $\mathbf{99.8}\%$ correct assignment.

The next object is a bust of Michelangelo's David shown in Figure 1a. The captured intersection network is given in Figure 1b. For this object, there are only two connected components. The first component contains only 23 points and spans the right arm whereas the second component con-

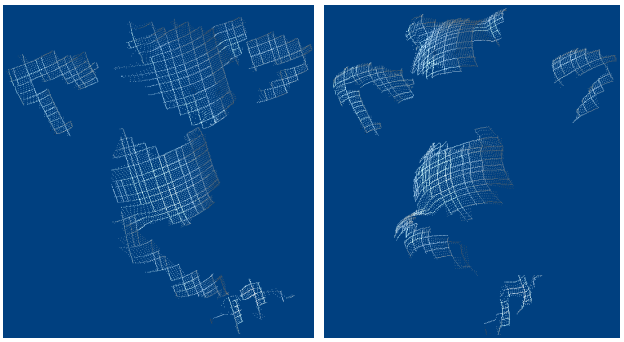


Figure 10: Two views of the reconstruction using the proposed method.

tains 1227 points and spans the body, head, the left arm and the background. There are 10 spurious connections inside the second component alone. These spurious connections link the head, the neck and the background. Since there is sufficient geometry around the spurious connections, all of them are solved correctly. In fact, for this object we have $\mathbf{100}\%$ correct assignment. The reconstructions are shown in Figures 1c and 1d.

As advocated, loopy belief propagation worked excellent in practice. Typically, it took about 10 iterations for convergence and we haven't experienced any cases where it did not converge. In unoptimized MATLAB code, the running time for a thousand grid intersections was about three minutes.

7. Conclusion and Future Work

In this paper, we have demonstrated a novel "one-shot" method to reconstruct 3D objects and scenes using a grid pattern. Unlike previous grid based approaches, our formulation, which uses a powerful probabilistic graphical model, can deal with scenes with complicated geometry. We have shown its performance in two such examples. The next step is to combine this system with an online registration method and to replace the DLP projector we currently use with a slide or laser projector in order to build a hand-held scanner.

8. Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0808718 and CCF-0729126. The authors would like to thank Erik B. Sudderth for many helpful discussions.

References

[1] J. Batlle, E. Mouaddib, and J. Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: A survey. *Pattern Recognition*, 1998.

[2] J. Y. Bouguet. Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calibdoc/index.html, 2001.

[3] B. J. Brown, C. Toler-Franklin, D. Nehab, M. Burns, D. Dobkin, A. Vlachopoulos, C. Doumas, S. Rusinkiewicz, and T. Weyrich. A system for high-volume acquisition and matching of fresco fragments: Reassembling Theran wall paintings. *ACM Transactions on Graphics SIGGRAPH*, 27(3), Aug. 2008.

[4] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20:470–480, 1998.

[5] H. Hiroshi Kawasaki, R. Furukawa, R. Sagawa, and Y. Yagi. Dynamic scene shape reconstruction using a single structured light pattern. *IEEE CVPR*, June 2008.

[6] S. Inokuchi, K. Sato, and F. Matsuda. Range imaging system for 3-d object recognition. In *IEEE ICIP*, 1984.

[7] T. Koninckx and L. Van Gool. Real-time range acquisition by adaptive structured light. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):432–445, March 2006.

[8] F. Kschischang, B. J. Frey, and H. andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. on Information Theory*, 47:498–519, 2001.

[9] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project: 3D scanning of large statues. In *ACM Transactions on Graphics SIGGRAPH*, pages 131–144, July 2000.

[10] J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, 1997.

[11] M. Proesmans, L. Van Gool, and A. Oosterlinck. Active acquisition of 3d shape for moving objects. *IEEE ICIP*, pages 647–650, 1996.

[12] G. Ru and G. Stockman. 3-d surface solution using structured light and constraint propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(4):390–402, 1989.

[13] J. Salvi, J. Batlle, and E. Mouaddib. A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters*, 19:1055–1065, 1998.

[14] J. Salvi, J. Pages, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, April 2004.

[15] E. B. Sudderth and W. Freeman. Signal and image processing with belief propagation. *Signal Processing Magazine, IEEE*, 25:114–141, March 2008.

[16] A. O. Ulusoy, F. Calakli, and G. Taubin. One-shot scanning using de bruijn spaced grids. In *Proceedings of the 3-D Digital Imaging and Modeling (3DIM)*, 2009.

[17] Y. Weiss and W. T. Freeman. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. *IEEE Trans. on Information Theory*, 47(2):736–744, 2001.

[18] L. Zhang, B. Curless, and S. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. *IEEE 3DPVT*, 2002.