One-Shot Scanning using De Bruijn Spaced Grids

Ali Osman Ulusoy* Fatih Calakli* Gabriel Taubin

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

Abstract

In this paper we present a new "one-shot" method to reconstruct the shape of dynamic 3D objects and scenes based on active illumination. In common with other related prior-art methods, a static grid pattern is projected onto the scene, a video sequence of the illuminated scene is captured, a shape estimate is produced independently for each video frame, and the one-shot property is realized at the expense of space resolution. The main challenge in grid-based one-shot methods is to engineer the pattern and algorithms so that the correspondence between pattern grid points and their images can be established very fast and without uncertainty. We present an efficient one-shot method which exploits simple geometric constraints to solve the correspondence problem. We also introduce De Bruijn spaced grids, a novel grid pattern, and show with strong empirical data that the resulting scheme is much more robust compared to those based on uniform spaced grids.

1. Introduction

The problem of estimating the shape of 3D objects and scenes from images has been studied since the early days of computer vision. A common approach to this problem is to use controlled illumination such as structured lighting, in which a projector is used to illuminate a 3D object with one or more special patterns and a camera is positioned to observe it. Most early works follow a temporal approach: multiple patterns are projected consecutively onto a static object and a camera captures an image for each projected pattern. These methods require the object not to move while the multiple patterns are being projected and the images being captured. They fail if the object moves or the scene changes during the sequence. To be able to scan *dynamic* objects and scenes, methods that recover depth information for each frame are needed. This property requires that the number of patterns projected be reduced to one *static* pattern, in which case all the information is transmitted in the spatial domain. These are the so-called *one-shot* methods, and the method introduced in this paper is one of these methods. We use a single grid pattern and present an efficient algorithm to recover depth information from a single camera image. We also propose a new pattern, which we call De Bruijn spaced grids and show empirically that it increases the robustness of reconstruction significantly. Figure 1b shows a typical input to our system and Figure 1c shows 3D reconstruction of the object.

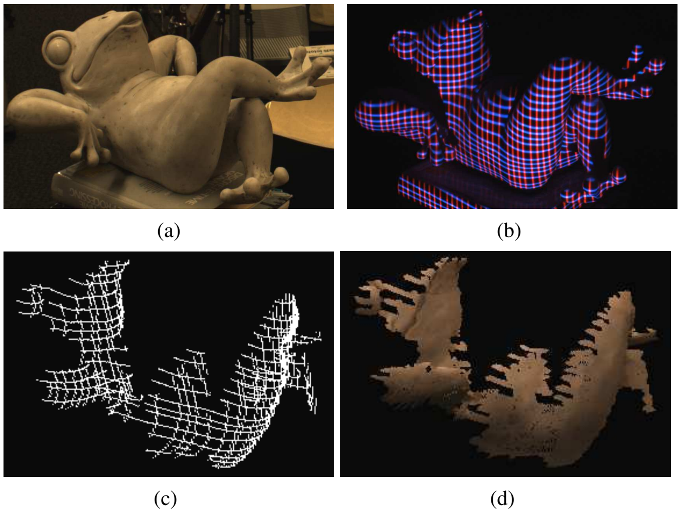

(a) (b) (c) (d)

Figure 1: (a) A typical 3D object, (b) input image, (c) 3D reconstruction of the object, (d) 3D reconstruction after texture mapping.

2. Related Work

Since a high variety of structured light approaches have been proposed to this day, we will only discuss some of the most relevant ones here. A comprehensive assessment is presented in [12, 1] by Salvi *et al.*

One of the first temporal structured light systems was

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

proposed by Posdamer and Altschuler [7], where the column number for projector columns are encoded in binary and consecutive patterns are projected over time, each corresponding to a bit plane. Note that to resolve M columns, one needs to project $logM$ patterns. In [6], Inokuchi *et al.* improve this by suggesting Gray codes which are more robust to decoding errors compared to simple binary encoding. More recently, Zhang and Huang demonstrated that real time reconstruction of dynamic shapes using a temporal approach is indeed possible at the expense of special hardware [15]. Note that here, we aim to propose a system using only off the shelf equipment such as a typical DLP projector and a color camera.

There has also been much work in purely spatial approaches. Zhang *et al.* [14] use a sequence of stripes with different colors. The transitions between colors in the sequence is a De Bruijn sequence which has the *windowed uniqueness* property, i.e. each window of size n in the sequence is unique. Other works include M-arrays [13], [12] which are two dimensional codes that have similar properties. Most of these approaches are highly sensitive to noise and require sophisticated image processing as they use complex illumination patterns. Furthermore, most of them rely heavily on color and therefore require accurate color calibration [3].

Ru and Stockman [10] suggest using a grid pattern (a binary pattern consisting of vertical and horizontal lines) and show that the problem is simplified greatly. The observed grid in the camera is matched to the projected pattern by exploiting a number of geometric and topological constraints and global 2D optimization. A major drawback of the approach 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. Proesmans *et al.* [9, 8] also use a grid pattern however, present a more efficient solution. Similarly, the algorithm employs relative numbering of the grid pattern and requires that most of the pattern be recovered well.

Recently, Kawasaki *et al.* [5, 4] proposed a colored grid pattern where vertical and horizontal lines are of different colors. Unlike most other grid based 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 performing singular value decomposition (SVD) on a large and very sparse matrix, which is an expensive operation and may be numerically inaccurate. Moreover, the algorithm may fail to converge to the correct reconstruction in some cases due to instabilities discussed in [5]. The authors suggest irregularly spaced grid patterns, which indeed increases the stability, however, does not guarantee *correctness*.

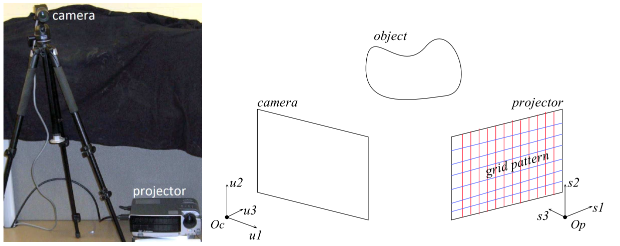


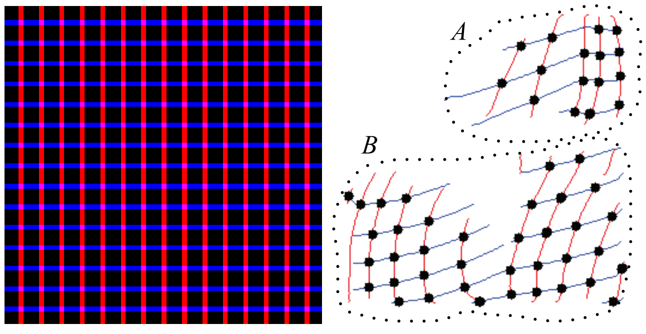
Figure 2: The system setup

In this paper, we use a grid pattern similar to the one used by Kawasaki *et al.*, with horizontal and vertical lines of different colors, but to simplify the search for correspondences between projected pattern crossings and pixels in the captured images we present an efficient algorithm performing safe propagation of correspondences by exploiting very simple geometric constraints, followed by a 1D search. The algorithm does not assume local smoothness and can handle undetected grid points or lost patches of the grid, a deficiency in most grid based approaches. Furthermore, we introduce De Bruijn spaced grids, which are grid patterns with spacings that follow a De Bruijn sequence, to make the search much more robust compared to using uniformly spaced grid patterns. This assures the stable and correct reconstruction of even small patches of grid points. Our novel formulation, which in addition to the De Bruijn spaced grid, significantly reduce the cost of searching for correspondences, results in an implementation which produces results of quality comparable to what the state of the art one-shot scanning methods produce.

3. Overview of the Approach

In this paper, we present a structured light approach consisting of a data projector and a single color camera, as shown in Figure 2. Both of which are calibrated with respect to a world coordinate system. The projector projects a known static grid pattern onto the 3D surface. Since the projected pattern is static, we could also use a much simpler and compact slide projector. An image of the object illuminated by the projected grid and captured by the camera is used to determine the depth information of points illuminated by grid crossings and observed by the camera. Producing the correct depths depend on being able to solve the so-called *correspondence* problem. Matching pairs of features in the camera and projector images need to be identified, as in the perspective of stereo vision [10]. Note that after correspondences are established, triangulation is trivial and depth information can be obtained easily.

An exhaustive search for the correspondences of grid points observed by the camera is intractable. However, using a pattern which disclosures spatial neighbor information for all the feature points, *e.g. a grid pattern*, the correspondence problem for the whole grid is simplified to identify-



(a) (b)

Figure 3: (a) A grid pattern, (b) two separate grid networks composed of captured intersections.

ing the correspondence for a single grid point, i.e. all grid points in a network can be uniquely identified based on a single correspondence. This can be done efficiently without assuming relative ordering of grid points and by exploiting simple geometric constraints. This will be explained in detail in Section 4.2.

At first, finding a single correspondence might seem to require a search through the whole grid points in the projector image. However, exploiting the *epipolar* constraints, the search space reduces to a single line. It shrinks even further because we know the points could have come from only grid crossings along the epipolar line.

Thus, using the described approach, the problem of recovering depth information is made as simple as obtaining a correspondence between a detected point (grid crossing) in the camera image and projected grid crossings along the point's epipolar line.

4. Reconstruction using a grid pattern

In the proposed system, the camera and the projector are assumed to be calibrated. The projector projects a grid pattern, composed of horizontal blue lines and vertical red lines as shown in Figure 3a, onto the 3D object and the camera is placed to capture the scene. A simple image segmentation process in the captured image, *e.g.*, *thresholding color channels*, is used to differentiate between the horizontal and vertical stripes.

The stripes detected are combined to form an input to the system - *a 2D grid network*. However, they, in general, may not result in a connected grid network due to shadows, occlusions and sharp depth discontinuities. The grid network is typically composed of several connected components. An example depicting this is given in Figure 3b. Our approach solves the correspondence problem for each grid network connected component independently. In this section, we only explain the steps of our method for solving a single connected grid network. Without loss of generality, the same steps can be applied to all grid networks independently for a complete 3D surface solution. The steps of the proposed approach is as follows:

1. Pick a grid point in the network. Using its epipolar line, find possible candidates for correspondence in the projector image.
2. For each candidate, propagate the correspondence to all grid points in the network. (This step provides a set of candidate solutions for a given 2D grid network.)
3. Choose the solution which matches the projected grid best via a 1D search

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

4.1. Finding candidates for correspondence

A grid pattern reduces the correspondence problem to the identification of a single correspondence. Nevertheless, the search for a single correspondence may still be costly considering the high number of grid points in the projector image. This motivates us to exploit geometrical relations between the camera and projector images to reduce the search space.

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 $\mathbf{u} = (u_1, u_2, 1)^t$ is

$$\lambda \mathbf{u} = \mathbf{p} \tag{1}$$

We define another coordinate system at the optical center of the projector for convenience and denote projector image point as $\mathbf{s} = (s_1, s_2, 1)^t$. 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.

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.

Equation (3) says that a captured grid point $\mathbf{u}$ in the camera image may correspond only to grid points on the epipolar line $L(\mathbf{u})$ in the projector image as shown in Figure 4.

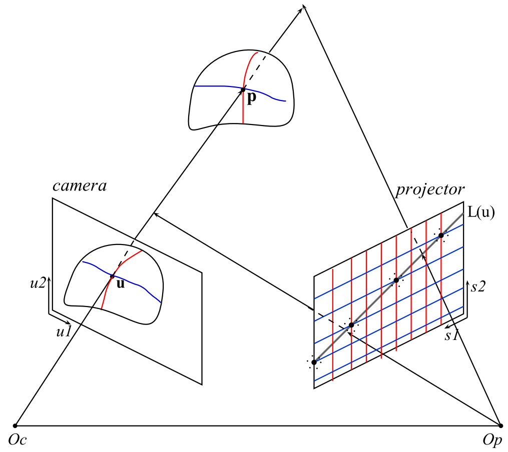


Figure 4: A point $\mathbf{u}$ and its epipolar line $L(\mathbf{u})$

The grid points on the epipolar line $L(\mathbf{u})$ provide a set of possible candidates for correspondence for a given 2D grid network.

4.2. Propagating a correspondence

Having obtained a list of correspondence candidates for a grid point, we show that the rest of the points can be inferred quite efficiently and without assuming relative ordering.

We begin by assuming that $\mathbf{u}$ came from a grid point $\mathbf{s}$ which satisfies $L(\mathbf{u})$. Note that $\mathbf{s}$ defines a single horizontal line and a vertical line on the projector plane as seen in Figure 5. Subsequently, these lines and O_p define horizontal and vertical planes $\Pi_h(\mathbf{s})$ and $\Pi_v(\mathbf{s})$ in the projector coordinate system as follows:

$$\Pi_h(\mathbf{s}) = \{\mathbf{p} : n_h(\mathbf{s}) \cdot (R\mathbf{p} + T) = 0\} \qquad (4)$$

$$\Pi_v(\mathbf{s}) = \{\mathbf{p} : n_v(\mathbf{s}) \cdot (R\mathbf{p} + T) = 0\} \qquad (5)$$

where the normals are defined as $n_h(\mathbf{s}) = [0 \ -1 \ \mathbf{s}_1]^t$ and $n_v(\mathbf{s}) = [-1 \ 0 \ \mathbf{s}_2]^t$.

We denote grid points horizontally linked to $\mathbf{u}$ as $u_{h-neighbors}$ and points vertically linked as $\mathbf{u}_{v-neighbors}$. Note that both $u_{h-neighbors}$ and $\mathbf{u}_{v-neighbors}$ are lists of grid points all linked to $\mathbf{u}$.

An important observation is that $\mathbf{u}_{h-neighbors}$ must have come from $\Pi_h(\mathbf{s})$ and $\mathbf{u}_{v-neighbors}$ from $\Pi_v(\mathbf{s})$. In other words, we know $\mathbf{u}_{h-neighbors}$'s row correspondence and $\mathbf{u}_{v-neighbors}$'s column correspondence. Given these, we can now compute $\mathbf{u}_{h-neighbors}$'s vertical correspondence and $\mathbf{u}_{v-neighbors}$'s horizontal correspondence easily by using 2, 4 and 5. Geometrically, this corresponds to intersecting the epipolar line of the neighbor point with its known correspondence, which is either a vertical or a horizontal line in the projector image plane. An example is given in Figure 6.

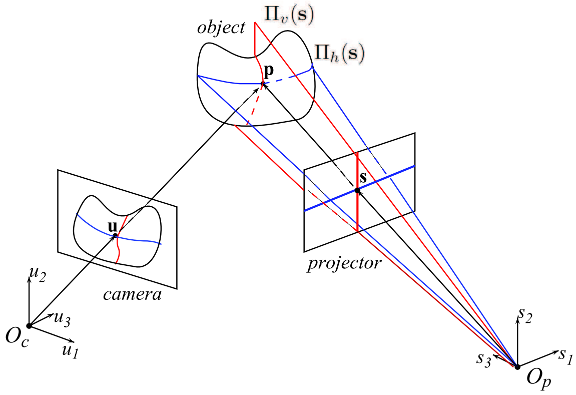


Figure 5: Two planes $\Pi_h(\mathbf{s})$ and $\Pi_v(\mathbf{s})$ defined by point $\mathbf{u}$'s correspondence $\mathbf{s}$.

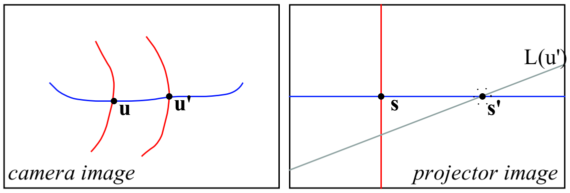


Figure 6: Let $\mathbf{u}' \in \mathbf{u}_{h-neighbors}$. Intersecting the $L(\mathbf{u}')$ with the horizontal line yields the vertical correspondence.

As a result, we obtain both row and column correspondences for $\mathbf{u}$'s neighbors. For points that are not $\mathbf{u}$'s horizontal or vertical neighbors; once the correspondences for the neighbors have been computed, they can propagate their solution to their own neighbors. An efficient algorithm that traverses each grid point at most once and guarantees that every point in the grid network is given a correspondence is presented in Algorithm 1.

4.3. Search for Correspondence

The propagation provides a set of solution candidates for a given grid network. As a final step, we need to identify the correct solution through the comparison of computed grid points $\mathbf{s}' \in S'$, and the projected grid points $\mathbf{s} \in S$ on the projector image, where S' is the set of computed grid points, and similarly S is the set of projected grid points.

Ideally, for the correct correspondence, the computed grid points $\mathbf{s}'$ must appear exactly on the projected (discrete) grid points $\mathbf{s}$. However, this, in general, may not be satisfied due to imperfections in calibration and image processing. Provided that the noise level is not excessively high, the location of a computed grid point $\mathbf{s}'$ still provides a good neighborhood which includes the true correspondence $\mathbf{s}$. Considering a computed grid point could move along any direction due to 2D noise, we assume that $\mathbf{s}'$ came from one of its four closest discrete neighbors $\mathbf{s}_{neighbors}(\mathbf{s}')$. The error function is defined to be the minimum cost of choosing either one of these neighbors. We expect correct choice $\mathbf{s}$ to

Algorithm 1 The Propagation Algorithm

```
{The correspondence for $\mathbf{u}$ denoted as $(\pi_v(u), \pi_h(u))$ is assumed to be $\mathbf{s}$.}
$\pi_h(u) \leftarrow \Pi_h(\mathbf{s})$, $\pi_v(u) \leftarrow \Pi_v(\mathbf{s})$
$Q \leftarrow \mathbf{u}$ {Initialize queue $Q$ with $\mathbf{u}$}
while $Q$ is not empty do
    $\mathbf{e} \leftarrow Q[1]$ {Get the first element from $Q$}
    if $\pi_v(\mathbf{e})$ exists then
        Compute $\pi_h(\mathbf{e})$
        $\pi_h(\mathbf{e}_{h-neighbors}) \leftarrow \pi_h(\mathbf{e})$
        Remove $(Q \cap \mathbf{e}_{h-neighbors})$ from $Q$
        Add $(\mathbf{e}_{h-neighbors} \setminus Q)$ to $Q$
    end if
    if $\pi_h(\mathbf{e})$ exists then
        Compute $\pi_v(\mathbf{e})$
        $\pi_v(\mathbf{e}_{v-neighbors}) \leftarrow \pi_v(\mathbf{e})$
        Remove $(Q \cap \mathbf{e}_{v-neighbors})$ from $Q$
        Add $(\mathbf{e}_{v-neighbors} \setminus Q)$ to $Q$
    end if
end while
```

satisfy the corresponding epipolar line of $\mathbf{u}'$. Note that $\mathbf{u}'$ is the camera image point corresponding to the computed projector image point $\mathbf{s}'$. Thus, a single cost is

$$dist(\mathbf{s}', \mathbf{s}_{neighbors}(\mathbf{s}')) = \min_{\mathbf{s} \in \mathbf{s}_{neighbors}(\mathbf{s}')} (|L(\mathbf{u}') \cdot \mathbf{s}|) \quad (6)$$

which is defined to be the minimum Euclidean distance of a discrete neighbor to the epipolar line $L(\mathbf{u}')$. Summing this distance metric for all the intersections, we get the error function

$$E(\mathbf{u}, \mathbf{s}) = \sum_{\mathbf{s}' \in S'} dist(\mathbf{s}', \mathbf{s}_{neighbors}(\mathbf{s}')) \quad (7)$$

which changes for various candidates $\mathbf{s}$ along the epipolar line $L(\mathbf{u})$. We choose the solution which minimizes the error function.

5. De Bruijn Spaced Grids

As described in 4.3, the minimum of the energy function (7) is taken to be the solution. Thus, the shape of this function affects to the robustness of the search. When using a regularly spaced grid, the function may not be sharply peaked, i.e. may have multiple local minima close to the global minimum. The error function for a small size set's (40 intersections) is given in Figure 7a. It is clearly seen that for small patches, using uniformly spaced grids result in a multiple significant peaks, which leads to unresolved ambiguities. In the presence of noise the search may result in a wrong correspondence. An example is depicted in Figure 7b.

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*.

We suggest grid spacings that follow a De Bruijn sequence to assure correct convergence. 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. There are multiple De Bruijn sequences for a given (k, n), each of length k^n. The choices of (k, n) were chosen empirically and are discussed in Section 6.

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 it "theoretically impossible" for wrong correspondence candidates to give false alarms and assures that the correct candidate be the global minimum. Of course, in the presence of calibration and image processing errors, one would expect to see some insignificant local minima.

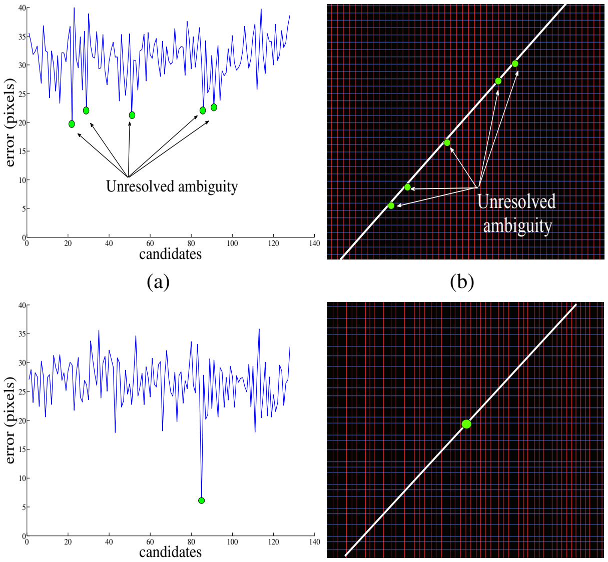
An example error function (for a set containing 40 intersections) using De Bruijn spaced grids ($n = 5$, $k = 3$) is given in Figure 7c. It can be observed that using De Bruijn spaced grids yielded a much more robust error function, resulting in a confident decision for making the correspondence. The correct correspondence is depicted along the epipole in Figure 7d.

6. Experiments and Results

Figure 2 shows the actual implementation of the scanner. We used a 1024 x 768 DLP projector and a 1600 x 1200 CCD camera. 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.

To recover the vertical and horizontal stripes, we simply threshold the respective color channels in the camera image. Then, we apply thinning to get the curves running from the center of each stripe. Finally, we find individual grid networks using connected component analysis. Note that this may yield multiple connected networks as shown in Figure 3b, which are handled independently as explained in Section 4.

To obtain the De Bruijn sequences, we used an online generator [11]. To decide on the parameters (k, n), consider the two extreme cases: when k is large and n is small and vice versa. In the first case, there will be many different spacings which will eventually lead to large spacings since the spacing alphabet is discrete. Thus, the reconstruction



(a) (b)

(c) (d)

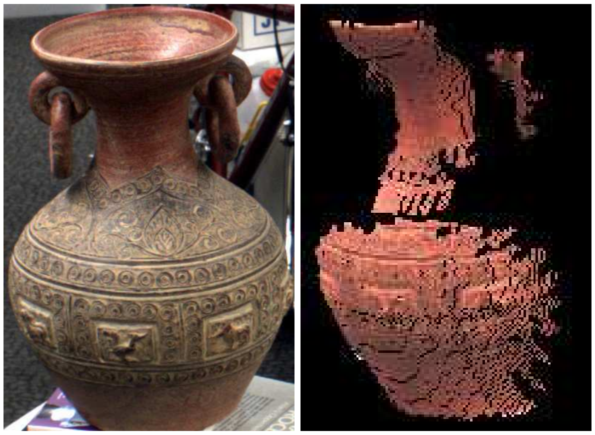
Figure 7: (a),(c) Typical values of the error function when using uniformly spaced and De Bruijn spaced grids respectively. Note the significant peak in 7c, (b),(d) the epipoles of the points considered and possible solutions are identified with green dots for (a) and (b) respectively.

will get sparse. However, even a very small patch of the pattern can easily be recovered since n is small. On the other extreme, when k is small and n is large, the pattern will be composed of a few large unique patches. Thus, reconstruction will require a small number of patches be identified since only a few can be enough to cover a whole surface. Note that one may not be able to reconstruct a surface if the detected patches are smaller than the needed size. This decreases the robustness to noise, depth discontinuities and texture.

In our experiments, we want to stay robust and still obtain a dense reconstruction. Empirically, we saw that $k = 5$ and $n = 3$ gave good results 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.

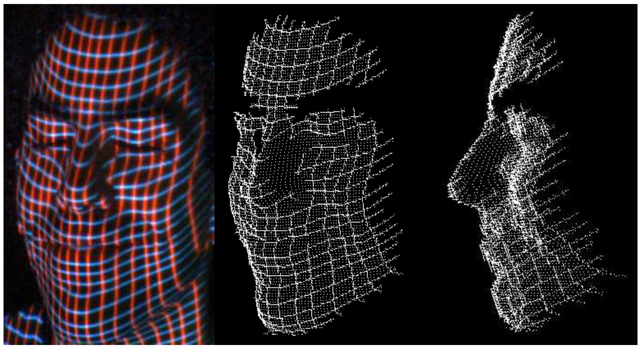
The results for a static textured object is given in Figure 8. It can be seen that although the object has both color and texture, it has been reconstructed very well. Another static object example is given in Figure 1. Since the object (Figure 1a) has sharp depth discontinuities and a lot of self occlusion, five independent grid networks were extracted. All of them were solved correctly as seen in Figure 1c.

As noted earlier, this algorithm can reconstruct dynamic shapes and scenes. To test this, we projected our static pattern and took multiple images as the shapes moved/deformed. Figure 9 shows results from a video of a human face and Figure 10 shows results from a sequence in which a hand moves.



(a) (b)

Figure 8: Reconstruction results for a textured object. (a) a pot, (b) the reconstruction after texture mapping.



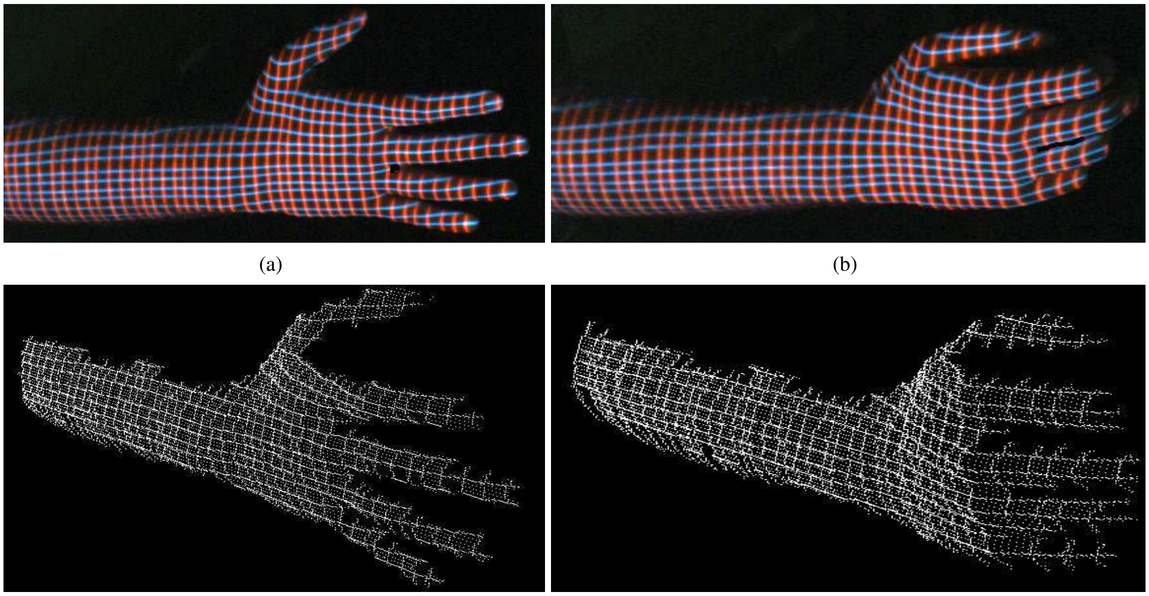
(a) (b) (c)

Figure 9: Reconstruction of a person's face. (a) a smiling human face, (b),(c) the reconstruction of the face.

7. Future Work and Conclusion

In this paper, we have demonstrated a novel "one-shot" method to reconstruct 3D objects and scenes using a grid pattern. Unlike most other grid based approaches, our algorithm is robust to most forms of texture, color and depth discontinuities and does not require sophisticated image processing. Furthermore, it is efficient as it exploits simple geometric constraints, employs fast traversal of the grid pattern and also avoids global 2D optimization. We have also introduced De Bruijn spaced grid patterns, which are grids with spacings that follow a De Bruijn sequence. We have shown with strong empirical data that the resulting scheme is much more robust compared to those based on uniform spaced grids.

A drawback of our approach, in fact all grid based approaches, is spurious connections in sharp depth discontinu-



(a)

(b)

(c)

(d)

Figure 10: Reconstruction of a person's hand as it closes. (a) an open human hand, (b) a closing human hand, (c) the reconstruction of the open hand, (d) the reconstruction of the closing hand.

ities. We plan to solve this issue in the future by traversing multiple paths to propagate solutions and comparing these solutions to detect and remove wrong links. We also plan to search for theoretical explanations and results regarding De Bruijn spaced grid patterns. Finally, we would like to implement this algorithm in real time as it shows great potential for GPGPU implementation.

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 Douglas Lanman for his assistance in projector calibration.

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/calib_doc/`, 2008.

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

[4] R. Furukawa, H. Q. H. Viet, H. Kawasaki, R. Sagawa, and Y. Yagi. One-shot range scanner using coplanarity constraints. *IEEE ICIP*, 2008.

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

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

[7] J. Posdamer and M. Altschuler. Surface measurement by space-encoded projected beam system. *Comput. Graphics Image Processing*, 18:1–17, 1982.

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

[9] M. Proesmans, L. Van Gool, and A. Oosterlinck. One-shot active 3d shape acquisition. *IEEE ICPR*, 1996.

[10] 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.

[11] F. Ruskey. The (combinatorial) object server. `http://theory.cs.uvic.ca/`, 2002.

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

[13] P. Vuylsteke and A. Oosterlinck. Range image acquisition with a single binary-encoded light pattern. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(2):148–164, Feb 1990.

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

[15] S. Zhang and P. Huang. High-resolution, real-time 3d shape acquisition. *IEEE CVPR*, 2004.