Structured Light Based Reconstruction Under Local Spatial Coherence Assumption

Hao Li, Raphael Straub, and Hartmut Prautzsch
Universität Karlsruhe (TH)
Karlsruhe, Germany
{hao,raphael,prau}@ira.uka.de

Abstract

3D scanning techniques based on structured light usually achieve robustness against outliers by performing multiple projections to simplify correspondence. However, for cases such as dynamic scenes, the number of frames captured from a certain view must be kept as low as possible, which makes it difficult to reconstruct complex scenes with high frequency shapes and inappropriate reflection properties. To tackle this problem, we present a novel set of color stripe patterns and a robust correspondence algorithm that assume local spatial coherence in the captured data. This assumption allows us to design our stripe sequences with globally unique neighborhood properties to effectively avoid wrong correspondences. The concept of local spatial coherence is further exploited to make the ensuing surface reconstruction practically insensitive to noise, outliers, and anisotropic sampling density. Thus, the recovery of a topologically consistent manifold surface can be drastically simplified. We have successfully generated high quality meshes of various colored objects using a minimalistic projector-camera system. In particular, the full sampling capabilities of our devices can be exhausted by taking only three shots.

1. Introduction

A projector and a camera form the basic requirements for building a 3D scanner based on structured light (cf. Fig. 1). In order to capture the shape of a real-world object, a sequence of patterns is projected onto the scene and their reflections are captured from a shifted position over time by taking photographs. Depending on where the pattern reflections are detected in the acquisitions, depth information can be accurately retrieved. This type of range measurement is called *active optical triangulation* [2] and typically yields a *point cloud* representing the object's surface.

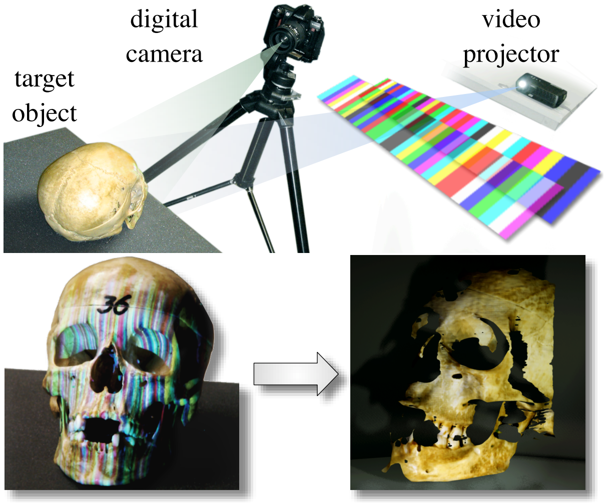


Figure 1. Top: Our minimalistic 3D scanner based on time-coded color structured light. Bottom: A textured reconstruction of a stripe illuminated skull.

Projecting a pattern with multiple stripes allows a two dimensional sampling of depth from a single input image and a faster acquisition. Depending on the projected stripe width and the acquisition resolution, the sampling densities along and between the stripe boundaries are generally different. This is particularly noticeable when we use the full resolution of our digital camera (6 Mpixel), which is much higher than that of our video projector (1 Mpixel). Without prior knowledge on how the point cloud is generated, most *surface reconstruction* methods from *unorganized points* fail to find the desired topology from such *anisotropic samplings*. Since most of the available range scanners rely on rather modest camera resolutions, this issue has not yet been explicitly addressed. As we do not wish to sacrifice accuracy and resolution, we choose to take on this challenge.

When multiple stripes are simultaneously projected, it is necessary to match the acquired stripe boundaries to the corresponding emitted ones. In this work, we project color stripes where two stripe projections suffice to uniquely encode stripe boundaries for identification and to exhaust the full projector resolution (one pixel wide stripes). By assuming negligible motion within two frames, our method is suitable for dynamic scenes. This property of the scene is called *local temporal coherence*.

The key innovation of our method is that the assumption of *local spatial coherence* (cf. Section 3.1) is not only used to identify stripe boundaries as in [3, 9, 16, 18, 12], but also to effectively enhance robustness against outliers. Our patterns are designed with stripe boundaries with globally unique neighborhood properties that are essential to avoid wrong correspondences due to high frequency occlusions.

In our correspondence algorithm, robustness is greatly improved by exploiting the synergetic effect of combining colors located at the medial axis of each stripe, which are more reliable, and those around the stripe boundaries, which are less susceptible to spatial discontinuities and occlusions. This method partly shifts the correspondence problem between projected and acquired data to that of within acquired data, which is similar in stereo vision. Hence, we can more easily handle the issues of *chromatic aberration*, the *chicken-wire effect*, and *unfocused stripe boundaries*, which are caused by video projector limitations and the use of high resolution captures (cf. Fig. 2).

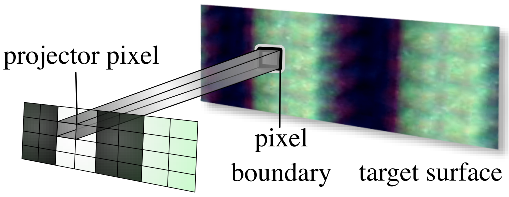
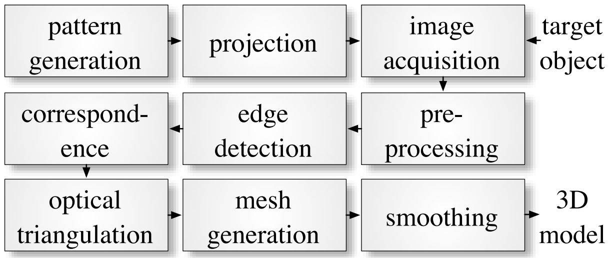


Figure 2. Pixel boundaries (chicken-wire effect) and color fringes around stripe boundaries (chromatic aberration) are visible due to limitations of the video projector.

Finally, we propose a surface reconstruction technique that is tightly integrated into the reconstruction pipeline. This involves extracting topology from the projected patterns and from the scanline ordering of the acquired images. It is therefore possible to make use of crucial a-priori knowledge of the underlying scan configuration to substantially simplify the task of generating a topologically consistent triangle mesh. In fact, this technique is insensitive to noise, outliers, and anisotropic sampling densities. Moreover, it also fills small undesired holes under the restriction of local spatial coherence.

2. Structured Light Techniques

A wide range of 3D scanning methods have been extensively investigated. An overview is presented by Curless et al. in [8] from which only active optical triangulation approaches based on structured light are the focus of our work. Our reconstruction pipeline follows a typical data flow:



According to Rusinkiewicz et al. [16], different structured light approaches can be distinguished by the underlying coherence assumptions of the scene. Methods that rely on an ideal scene reflectance (cf. [4, 6]) are too restrictive and unreliable for general usage. For this reason, only approaches that assume spatial and temporal coherences are relevant in this paper. Regardless of the application, the objective is to relax these assumptions by finding a good trade-off between speed, robustness, and accuracy.

Approaches based on *global temporal coherence* assume a static scene during the entire scanning. Posdamer and Altschuler introduced in [14] a set of binary coded stripe patterns that produced a sequence of n unique stripes by accumulating $\log n$ projections over time. Using *Gray codes*, Sato and Inokuchi [17] improved this approach by generating broader stripes for better detection. The same technique was adapted by Rocchini et al. [15], in which accuracy was ameliorated with uniform light intensity color stripes. To further increase speed, Caspi et al. generalized in [5] the concept of Gray codes by also using color encoding.

An effective way to reduce the number of input images is to consider spatial continuity in the scene. Koninckx et al. recently presented in [11] a one-shot technique based on global spatial coherence which allows objects with occlusions by using a self-adaptive pattern. The required adaptation process however disallows fast changes in the scene. Another avenue to reconstruct objects with occlusions is to assume piecewise continuity. One solution is to encode stripe boundaries with their adjacent stripe colors. In [3], only the three primary colors were used and the correspondence algorithm iteratively takes the longest subsequence of corresponded stripe boundaries. According to [18], this method has proven to be unsatisfactory, yielding many outliers and holes. Davies and Nixon embarked on a similar technique in [9] by projecting dots in a hexagonal arrangement at the cost of a more complex segmentation. In particular, two-dimensional local spatial coherence is assumed in the captured data as in our method. However, this is not the case in other techniques involving stripe projections.

One approach to improve robustness against high frequency discontinuities is to formulate the correspondence as a global optimization problem. This has been demonstrated by Zhang et al. in [18] using a multi-pass dynamic programming approach and a colored stripe pattern that is generated from a so-called *DeBruijn sequence*, which guarantees that very small subsequences are unique within the whole sequence. However, this approach exhibits flaws when longer stripe sequences are not captured by the camera and multiple optimal solutions are possible. In fact, the reconstructed mesh can be broken into several disconnected parts as observed in many examples in [12], especially around surface boundaries.

A more effective way to enhanced robustness is to use very few additional acquisitions assuming local temporal coherence. A remarkable example is the real-time acquisition technique proposed by Rusinkiewicz et al. in [16]. It requires four frames to encode a sequence of 111 unique stripe boundaries with additional spatio-temporal constraints to enhance robustness against slight motions. They define local spatial coherence by the minimum number of camera pixels (two in this case) required to identify a projected feature, which is resolution dependent. Because of the limited depth of field in video projectors, stripe boundaries would appear out of focus when the photographs have a higher resolution.

Our *spatio-temporal* approach uniquely encodes stripe boundaries via adjacent stripe colors and is able to exhaust the full projector resolution with two stripe projections only. It is independent of the input image resolution and can achieve higher robustness against outliers.

3. Range Acquisition

This section explains how to generate a point cloud that represents the reconstructed surface from a set of input photographs. The first step consists of projecting our stripe patterns onto the scene to perform active optical triangulation. A diligent design of the patterns and an algorithm for generating them are described in Section 3.1.

During the acquisition, the scene is only illuminated by the video projector in order to avoid intervention from other light sources. An additional photograph of a plain white projected scene is used for foreground segmentation and color correction as in [12]. It is worth highlighting that this supplementary shot opens up the possibility for texture reconstruction (cf. Fig. 1). On each shot of the stripe illuminated scene, we apply a scalar median filter in each RGB-channel independently to reduce noise that is mainly caused by the chicken-wire effect (cf. Fig. 2).

For the optical triangulation, stripe boundaries must be localized on the camera image plane. This is done by simple color edge detection on each scanline as described in [18, 12]. To obtain subpixel accuracy, we use the same linear interpolation technique presented in [13]. Diminishing false positive edges that do not belong to the projected stripe boundaries is possible via a non-maxima suppression method suggested in [12]. According to our designed stripe patterns, some stripe boundaries might not be present in all frames (cf. Section 3.1). For this reason, detected edges from all acquired frames are merged into one data set.

Our correspondence algorithm then matches the sequence of acquired stripe boundaries with the projected ones for each row. This is presented at length in Section 3.2. Next, we eliminate false positive stripe boundaries that do not belong to our projected stripe patterns using the properties described in Section 3.1. Due to our assumption of local spatial coherence, a stripe boundary is considered valid if at least one of its acquired neighbors is also a neighbor stripe boundary in the projected pattern. By removing invalid correspondences, outliers can be significantly reduced at the cost of some eliminated correct data. Finally, optical triangulation is performed to obtain the desired point cloud as illustrated in Figure 7(c).

3.1. A Spatio-temporal Pattern

The design of a structured light pattern depends on spatial and temporal coherence assumptions. Rusinkiewicz et al. [16] assume that two adjacent camera pixels observe the same spot on the surface and refer to it as *local spatial coherence*. Here, we use vertical stripes and assume that in general two adjacent stripe boundaries in a camera scanline correspond to adjacent projected stripe boundaries. We call this property *horizontal spatial coherence*. Additionally, we assume that stripe boundaries detected in any camera scanline are also present in the adjacent scanlines in general. We call this *vertical spatial coherence*. Note that these assumptions are independent of the camera resolution. We will exploit this in Sections 3.2 and 4.

In order to more reliably acquire scenes, we restrict the number of colors in the projected pattern to the RGB-colors $\mathbf{c} \in \{0,1\}^3$, where $[1 \quad 1 \quad 1]^{\mathrm{t}}$ represents white, for instance. Let $\mathbf{P} = [\mathbf{p}_1 \ldots \mathbf{p}_n]$ represent our projected spatio-temporal stripe sequence. By projecting t stripe patterns sequentially, the accumulation of t projected colors of a single stripe form one of 2^{3t} possible color combinations $\mathbf{p}_i = [\mathbf{c}_i^1 \ldots \mathbf{c}_i^t]$, where each $\mathbf{c}_i^j \in \{0,1\}^3$ represents a stripe color of a specific frame.

We uniquely encode each stripe boundary by its two adjacent stripe color combinations and obtain the stripe boundary sequence $\mathbf{Q} = [\mathbf{q}_1 \ldots \mathbf{q}_{n-1}]$, where $\mathbf{q}_i =$

$[\mathbf{p}_i\mathbf{p}_{i+1}]$ represents a single stripe boundary. This is illustrated in Figure 3.

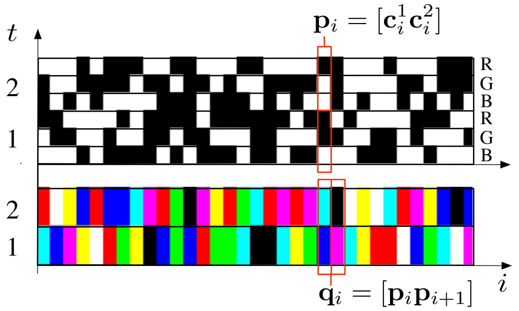

$\mathbf{p}_i = [\mathbf{c}_i^1\mathbf{c}_i^2]$
t
2
1
R
G
B
R
G
B
2
1
$\mathbf{q}_i = [\mathbf{p}_i\mathbf{p}_{i+1}]$
i

Figure 3. Color-coded stripes $\mathbf{p}_i$ and the corresponding stripe boundaries $\mathbf{q}_i$.

A stripe boundary is only detectable if it has a color transition in at least one of the projections:

$$\forall i \in \{1, \ldots, n-1\} : \mathbf{p}_i \neq \mathbf{p}_{i+1} \,. \tag{1}$$

Thus, there are $2^{3t} \cdot (2^{3t} - 1)$ different stripe boundaries.

However, when only adjacent stripe colors are considered, local incoherences due to high frequency occlusions are likely to produce ambiguous correspondences. For instance, Figure 4 illustrates a situation where a projected stripe is not visible to the camera and a wrongly captured stripe boundary is identical to another projected one. The same problem occurs when some areas are not illuminated by the projector.

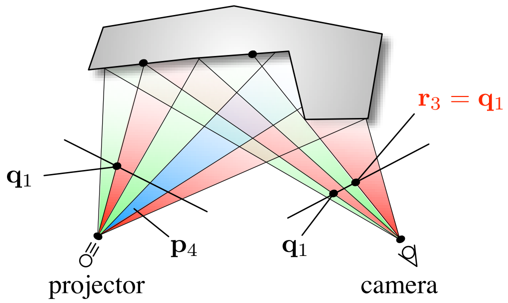


Figure 4. As the stripe $\mathbf{p}_4$ is completely occluded, the wrongly acquired stripe boundary $\mathbf{r}_3$ can be falsely corresponded to the projected stripe boundary $\mathbf{q}_1$.

Here, we alleviate misidentifications of this type by adding spatio-temporal constraints in our stripe patterns. Analogous to [16], we first ensure that each stripe boundary changes over time:

$$\begin{gathered} \forall i \in \{1, \ldots, n-1\} \exists t_1, t_2 : t_1 \neq t_2 \,, \\ [\mathbf{c}_i^{t_1}\mathbf{c}_{i+1}^{t_1}] \neq [\mathbf{c}_i^{t_2}\mathbf{c}_{i+1}^{t_2}] \,. \end{gathered} \tag{2}$$

This helps to distinguish between projected stripe boundaries and false positive detected ones.

However, the problem of newly emerged stripe boundaries, as illustrated in Figure 4, still remains. Such stripe boundaries can only be identified if they are not present in the projected pattern. As they emerge from two stripes in close vicinity, we avoid the problem by ensuring that stripe color combinations of a pair of close stripes are unique within all other close pairs. We express this as:

$$\begin{gathered} \forall i, j, k, l \in \{1, \ldots, n\}, \\ 1 \leq |i - j| \leq d, 1 \leq |k - l| \leq d : \\ [\mathbf{p}_i\mathbf{p}_j] = [\mathbf{p}_k\mathbf{p}_l] \;\Rightarrow\; (i = k) \wedge (j = l) \,, \end{gathered} \tag{3}$$

where two stripes are defined to be *close* if their distances are less than or equal to a predefined d. In this way, up to $d - 1$ consecutive missing stripes can be tolerated.

At this point, it is still unclear how to generate a stripe pattern with the above stated properties or whether such a pattern with a minimum sequence length n and closeness d exists. Answering this question is particularly difficult because Property (3) is global. We propose an algorithm that incrementally adds a new stripe to the sequence and ensures that the Properties (1), (2), and (3) are satisfied.

For fast verification of the global property, we use a *hash table* containing all close stripe pairs in the stripe pattern built so far. For each stripe, all stripe color combinations are consecutively tested. The order of choosing new stripe color combinations should be random as an exhaustive test of color combinations in the same order for each stripe would lead to repetitions and thus more failures. If none of the color combinations is valid, back tracking is required and another color combination is chosen for a previous stripe. If the algorithm terminates without any results, a stripe pattern with the neighborhood properties does not exist. The worst-case running time is $O(2^{3tn} \cdot d)$, assuming each hash table operation to be $O(1)$. Finding the optimal pattern with the maximum possible d for a given t and n would thus be intractable for large t and n. Nevertheless, suboptimal patterns with large enough d can be pre-computed within a reasonable amount of time as shown in Table 1.

3.2. Correspondence

Once the stripe boundaries are detected for each row, they must be correctly assigned to the projected ones. Although they are uniquely encoded with their adjacent stripe color combinations, pixel values in the vicinity of the acquired stripe boundaries are not suitable for identification,

$n_{\max}$		d			
		1	2	3	4
t	1	≥ 26	11	9	8
	2	≥ 1955	≥ 893	≥ 525	≥ 337
	3	≥ 130420	≥ 62253	≥ 37967	≥ 25711

$n_{\max}$		d			
		5	6	7	8
t	1	8	8	8	8
	2	≥ 235	≥ 165	≥ 134	≥ 93
	3	≥ 18488	≥ 13855	≥ 10647	≥ 8479

Table 1. Maximum possible number $n_{\max}$ of stripes satisfying Equations (1), (2), and (3) for a given number of acquired frames t and distance d.

especially in high resolution photographs. As noted earlier, this is mainly due to hardware limitations of the video projector. On the other hand, two color combinations that are measured at the medial axes of two adjacent stripes in a camera scanline are more discriminating and expressive than those around stripe boundaries—provided horizontal spatial coherence is present in the scene. However, these color combinations suffer from high frequency spatial discontinuities, undetected stripe boundaries, and overdetected ones. Conversely, color combinations obtained from the vicinity of stripe boundaries are almost unaffected by these problems, but are less expressive.

Assuming vertical spatial coherence, our method clusters stripe boundaries of similar color combinations across the scanlines. All stripe boundaries have both a left and a right side *medial axis color* and the medians of each side within a cluster are used for correspondence. Because of occlusions, neither the number of resulting clusters nor their corresponding representatives are known in advance.

Let $\mathbf{r}_{jk}$ be the kth acquired stripe boundary in row j:

$$\begin{gathered}\mathbf{r}_{jk} = [\mathbf{a}_{jk}\mathbf{a}_{j,k+1}] \in [0,1]^{6t} \ , \\ j = 1, \ldots, r \ , \ k = 1, \ldots, m_j \ ,\end{gathered}$$

where $\mathbf{a}_{jk}$ and $\mathbf{a}_{j,k+1}$ are the acquired color combinations of the respective left and right pixels adjacent to the stripe boundary. At the beginning, each first row stripe boundary $\mathbf{r}_{1k}$ belongs to a newly initialized cluster $\mathcal{C}_k$. Each cluster $\mathcal{C}_i$ is represented by its centroid $\mathbf{c}_i$:

$$\mathbf{c}_i = \frac{1}{|\mathcal{C}_i|} \sum_{\mathbf{r} \in \mathcal{C}_i} \mathbf{r} \in [0,1]^{6t} \ .$$

For the next row, we assign each $\mathbf{r}_{2k}$ to the cluster with the most similar centroid. If $\mathbf{r}_{2k}$ is too different from all $\mathbf{c}_i$, we initialize a new cluster for it. This is repeated for the remaining rows where each $\mathbf{c}_i$ is updated accordingly. At most one *feature vector* $\mathbf{r}_{jk}$ from a row j should be assigned to a specific cluster $\mathcal{C}_i$. Analogous to stereo vision correspondence, we call this restriction *uniqueness constraint*.

Each stripe boundary has a medial axis color on its left and right side. For each $\mathcal{C}_i$, we denote the set of all left side and right side medial axis colors by $\mathcal{S}_i^{\mathrm{l}} = \{\mathbf{s}_1^{(i),\mathrm{l}}, \ldots, \mathbf{s}_{q_i}^{(i),\mathrm{l}}\}$ and $\mathcal{S}_i^{\mathrm{r}} = \{\mathbf{s}_1^{(i),\mathrm{r}}, \ldots, \mathbf{s}_{q_i}^{(i),\mathrm{r}}\}$, respectively (cf. Fig. 5).

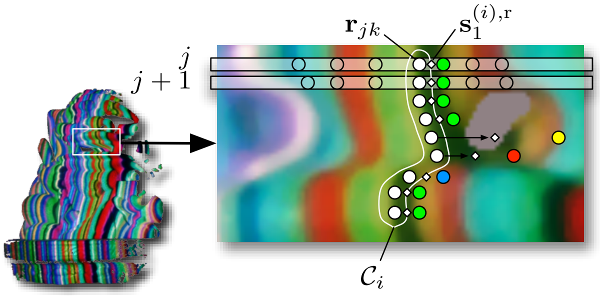


Figure 5. The clusters $\mathcal{C}_i$ are computed by inter-scanline clustering over the colors at the vicinity of the stripe boundaries. The medians of the color values $\mathrm{s}_l^{(i),\mathrm{l/r}}$ on the medial axis are used for the correspondence.

The left and right side medians of each cluster $\mathcal{C}_i$ are given by $\tilde{\mathrm{s}}_i := [\tilde{\mathrm{s}}_i^{\mathrm{l}}\tilde{\mathrm{s}}_i^{\mathrm{r}}]$, where $\tilde{\mathrm{s}}_i^{\mathrm{l}} := \operatorname{median} \mathcal{S}_i^{\mathrm{l}}$ and $\tilde{\mathrm{s}}_i^{\mathrm{r}} := \operatorname{median} \mathcal{S}_i^{\mathrm{r}}$. All detected stripe boundaries $\mathbf{r}_{jk} \in \mathcal{C}_i$ are assigned to $\tilde{\mathrm{s}}_i$ which is also denoted by $\tilde{\mathrm{s}}_{jk}$.

Let $\mathbf{Q} = [\mathbf{q}_1 \ldots \mathbf{q}_{n-1}]$ be the projected stripe boundary sequence. The aim is to match each $\mathbf{r}_{jk}$ with a $\mathbf{q}_i$ which is expressed by the identification map Φ_j, where $\Phi_j(k) = i$. For each row j, we iteratively determine for all k the closest pairs $(\tilde{\mathrm{s}}_{jk}, \mathbf{q}_i)$ until no more matches are possible. Whenever a match between two stripe boundaries is found, the uniqueness constraint disallows any further matches with one of these two stripe boundaries within the same row. Hence, Φ_j is an injective function.

A brute force approach has a running time of $O(r \cdot m_j \cdot n \cdot \min\{m_j, n\})$ (cf. Alg. 1). A more efficient implementation that requires only $O(r \cdot m_j \cdot n \cdot \log n)$ can be achieved by presorting the distances d_{ik}.

Although the proposed clustering is dependent on the scanline order, we manage to obtain a more robust correspondence for high resolution captures because of the two-dimensional local spatial coherence assumption. While in general, a globally suboptimal correspondence is computed and the point cloud obtained by optical triangulation is still affected by outliers (cf. Fig. 7(c)), the errors are more evenly distributed over the entire shape than using the dynamic pro-

Algorithm 1 correspondStripeBoundaries

```
for all $j = 1, \ldots, r$ do
    $P := \{1, \ldots, n-1\}$, $A := \{1, \ldots, m_j\}$
    for all $k = 1, \ldots, m_j$ do
        for all $i = 1, \ldots, n-1$ do
            $d_{ik} := \|\mathbf{q}_i - \tilde{\mathbf{s}}_{jk}\|$
    while $P \neq \emptyset \ \wedge \ A \neq \emptyset$ do
        $(i_{\min}, k_{\min}) := \mathrm{argmin}_{(i,k) \in P \times A}\, d_{ik}$
        $\Phi_j(k_{\min}) := i_{\min}$
        $P := P \setminus \{i_{\min}\}$, $A := A \setminus \{k_{\min}\}$
return $\Phi_1, \ldots, \Phi_r$
```

gramming approach presented by Zhang et al. in [18] as observed in [12]. This is particularly important as it simplifies the task of removing outliers for surface reconstruction.

4. Surface Reconstruction

After optical triangulation, the resulting point cloud is given by the set of points $\mathbf{x}_{jk} \in \mathbb{R}^3$ each corresponding to a detected stripe boundary $\mathbf{r}_{jk}$. The goal of our surface reconstruction is to approximate this point cloud with a 2-manifold triangular mesh that is free of outliers and noise.

Much research has been conducted on surface reconstruction from point clouds and a complete review would be beyond the scope of this paper. An excellent overview can be found in [1]. However, since we use high resolution captures, vertical sampling densities are much higher than the horizontal ones (cf. Fig. 7(c) and (d)) and using common methods would be doomed to fail. The same applies if the surface is extracted from *range scans* by computing a triangulation on the camera image plane as in [7].

For this reason, we simplify the problem by extracting useful a-priori information from the previous correspondence stage and the underlying projector-camera system:

- **topology** Two stripe boundaries $\mathbf{r}_{j,k_1}$ and $\mathbf{r}_{j+1,k_2}$ from two adjacent camera scanlines representing the same projected stripe boundary, i. e., $\Phi_j(k_1) = \Phi_{j+1}(k_2)$, are connected vertically. If two acquired neighboring stripe boundaries are also adjacent stripe boundaries in the projected pattern, they are connected horizontally.

- **spatial coherence** Meshing only with the above topology information would yield many holes in the reconstruction because of undetected stripe boundaries and eliminated correspondences. Holes that are not due to shadow areas can be filled using our spatial coherence assumptions.

- **orientation** A consistent orientation can be easily determined since the normals on the object's surface must always point towards the camera and the projector, of which the positions and orientations are known.

Our surface reconstruction algorithm requires two input parameters $\tau_{\mathrm{h}}, \tau_{\mathrm{v}} \in \mathbb{N}$. τ_{h} denotes the maximum number of consecutive stripe boundaries allowed to be missing in a scanline to represent a small horizontal gap. τ_{v} is the maximum number of consecutive rows in which a specific projected stripe boundary is not detected to represent a small vertical gap.

We begin by assuming horizontal spatial coherence for each scanline and fill small horizontal gaps so that only triangles between two adjacent stripe boundaries need to be generated. This is done by verifying if the widths of the gaps $w_{jk} := \Phi_j(k+1) - \Phi_j(k) - 1$ are less than or equal to τ_{h} for all $j = 1, \ldots, r$ and $k = 1, \ldots, m_j - 1$. If a small gap is found, w_{jk} points are added (one for each missing projected stripe boundary) and their positions are estimated by linearly interpolating between $\mathbf{x}_{jk}$ and $\mathbf{x}_{j,k+1}$. The point cloud and the correspondences Φ_j are then updated with the newly interpolated points and correspondences.

Let J_i be the ordered set of all camera scanlines in which the projected stripe boundary $\mathbf{q}_i$ is visible and let j_{pi} be the pth largest element in J_i. More precisely:

$$\begin{aligned} J_i \quad &:= \quad \{j \mid \Phi_j(k) = i\,,\, k \in \{1, \ldots, m_j\}\} \\ &= \quad \{j_{1i}, \ldots, j_{pi}, \ldots, j_{|J_i|,i}\}\,, \\ & \qquad \text{where } j_{1i} < \ldots < j_{pi} < \ldots < j_{|J_i|,i}\,. \end{aligned}$$

In particular, J_i and J_{i+1} originate from two adjacent projected stripe boundaries. For all $i = 1, \ldots, n-1$, our mesh generation algorithm iteratively creates triangle strips by vertically advancing to the next largest element from either J_i or J_{i+1} (cf. Fig. 6). Thus, at each iteration step, we distinguish between a left and a right triangle configuration with heights $h_l = j_{l+1,i} - \min\{j_{li}, j_{r,i+1}\}$ and $h_r = j_{r+1,i+1} - \min\{j_{li}, j_{r,i+1}\}$, respectively, which is counted in number of rows on the camera image plane. If both h_l and h_r are greater than τ_{v}, we assume a hole due to shadow areas and advance to the next smallest element of J_i or J_{i+1}, i. e., $j_{l+1,i}$ or $j_{r+1,i+1}$, whichever is the smaller. Otherwise, the triangle with the shortest height is taken. As shown in Figure 7(d), this method is insensitive to highly anisotropic sampled data.

However, the resulting mesh is still affected by outliers which can be easily eliminated once the mesh topology is found. Analogous to [7], we first remove triangles with very long edges. In addition, we leave out back-faced triangles and those with normals that are almost orthogonal to their lines of sight. Note that holes due to the removal of outliers must be refilled. In practice, mesh smoothing is necessary to reduce noise due to measurement inaccuracies. We adapted a standard Laplacian smoothing based on curvature flow as presented in [10].

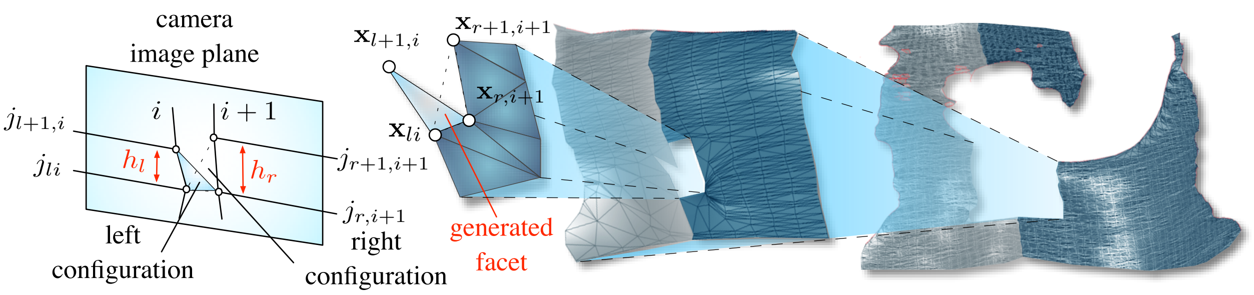


Figure 6. Triangle mesh generation from a highly anisotropic sampled point cloud.

5 Results

Our system consists of two low cost devices: an ASK M2 DLP video projector and a Fuji Finepix S2 Pro digital camera (cf. Fig. 1). To achieve the highest sampling resolution, we project stripes with one pixel width and take photographs with 3024×2016 pixels. This is in contrast to most other works which use resolutions far below 1 Mpixel [15, 16, 18, 11]. As we scan in complete darkness, the depth of field of the camera is increased by choosing a small aperture ($f/22$) and a longer exposure time (3 s). The full sampling capabilities of our system are exhausted with only three shots (two structured light and an additional plain white projection). Our proposed technique has therefore the potential to be an integral part of range scanners for dynamic scenes based on state-of-the-art technologies such as high definition video projectors and camcorders.

From our experiments, we conclude that the main hardware limitations are due to the video projector. For instance, the effects of chromatic aberration and chicken-wire seriously affected our acquisitions. And as with most active scanners using video projectors, keeping sharply focused stripe boundaries within a reasonable depth of field is hard.

Figure 7(a) and (b) demonstrate the performance of our reconstruction on two particularly challenging objects. While the human skull has relatively good reflectance properties, it is affected by many concavities and self-occlusions. Acquiring the shape of the guardian lion is also difficult because of its high frequency and dark granite texture, which yields many small holes in the reconstruction. The mesh of the skull has 41 k vertices and 79 k triangles and the lion 28 k vertices and 53 k triangles. Both shapes were acquired using patterns with 200 stripes and $d = 5$. Clustering as presented in Section 3.2 computed 549 partitions in 43 s (skull) and 818 partitions in 23 s (lion). The correspondences computed with presorted d_{ik} in Algorithm 1 took 1 min 25 s (skull) and 2 min 56 s (lion) instead of 9 min 42 s (skull) and 19 min 6 s (lion) without sorting. We used $\tau_\mathrm{h} = 2$ and $\tau_\mathrm{v} = 40$ for the surface reconstruction of both objects, which was followed by 20 iterations of Laplacian smoothing. Their complete reconstruction times were below 5 min and no additional manual post processing was required. All experiments were performed on a 1.8 GHz PowerPC 970 FX with 768 MB RAM.

Currently, we see two main areas of improvement in our approach. As our algorithm for stripe pattern generation has an exponential runtime, we are interested in finding a more efficient way to compute it. Another question would be how to efficiently perform the stripe boundary clustering without depending on the scanline order.

Acknowledgements

We would like to thank Yuanshan Lee for proof-reading this paper. The human skull was generously provided by Jin Li with the agreement of Marek Janczak from the University Medical Centre Heidelberg.

References

[1] F. Bernardini and H. E. Rushmeier. The 3d model acquisition pipeline. *Comput. Graph. Forum*, 21(2):149–172, 2002.

[2] P. J. Besl. Active, optical range imaging sensors. *Mach. Vision Appl.*, 1(2):127–152, 1988.

[3] K. L. Boyer and A. C. Kak. Color-encoded structured light for rapid active ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1):14–28, 1987.

[4] B. Carrihill and R. Hummel. Experiments with the intensity ratio depth sensor. *Computer Vision, Graphics, and Image Processing*, 32:337–358, 1985.

[5] D. Caspi, N. Kiryati, and J. Shamir. Range imaging with adaptive color structured light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):470–480, 1998.

[6] G. Chazan and N. Kiryati. Pyramidal intensity-ratio depth sensor. Technical Report 121, Center for Communication and Information Technologies, Department of Electrical Engineering, Technion, Haifa, Israel, 1995.

[7] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer*

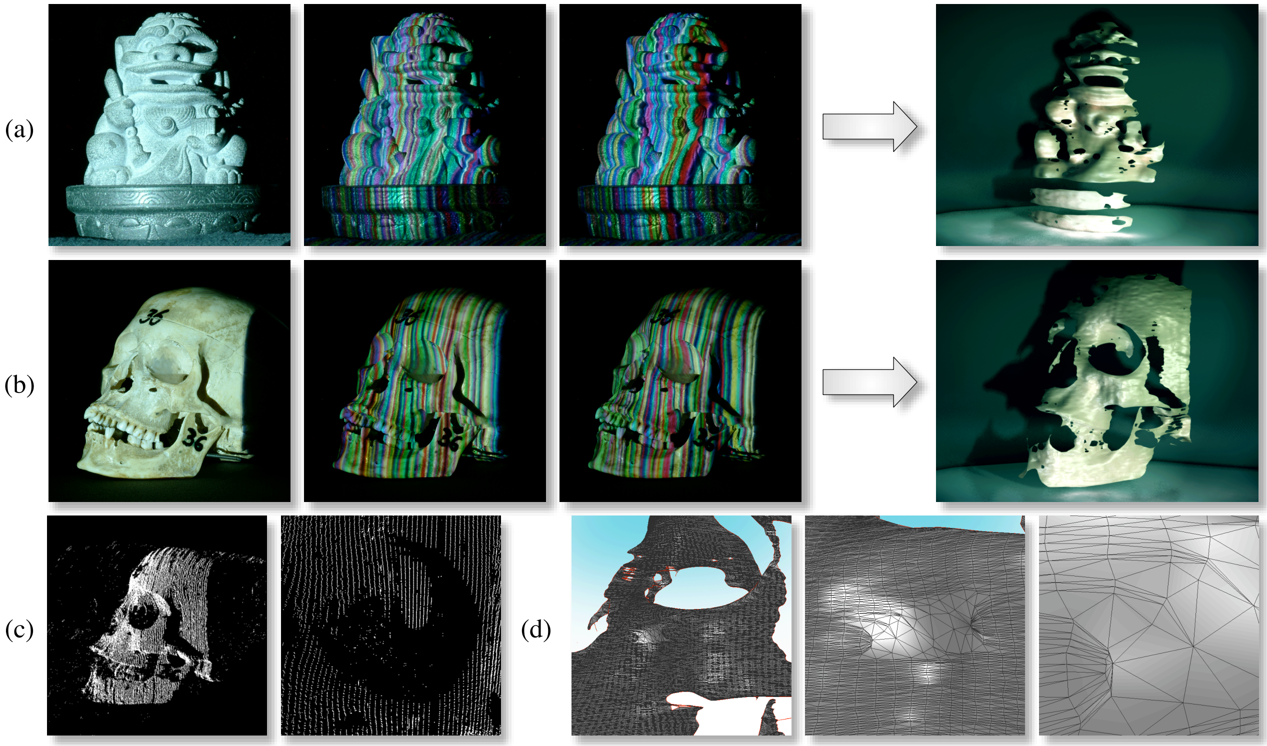


Figure 7. The reconstruction of the guardian lion (a) and the human skull (b) with three input images each shown on the left. The irregularly sampled point cloud from high resolution captures contains outliers and noise (c). Even with these severe errors our surface reconstruction algorithm is able to generate a consistent triangle mesh from this point cloud (d).

graphics and interactive techniques, pages 303–312, New York, NY, USA, 1996. ACM Press.

[8] B. Curless, S. Seitz, J.-Y. Bouguet, P. Debevec, M. Levoy, and S. Nayar. 3d photography. In *ACM Siggraph 2000 Course Notes, Computer Graphics Proceedings, Course No. 19*, August 24th 2000.

[9] C. J. Davies and M. S. Nixon. A hough transform for detecting the location and orientation of three-dimensional surfaces via color encoded spots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(1):90–95, 1998.

[10] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[11] T. P. Koninckx, A. Griesser, and L. J. V. Gool. Real-time range scanning of deformable surfaces by adaptively coded structured light. In *3DIM*, pages 293–301, 2003.

[12] H. Li, R. Straub, and H. Prautzsch. Fast subpixel accurate reconstruction using color structured light. In *Proceedings of the Fourth IASTED International Conference on Visualization, Imaging, and Image Processing*, pages 396–401. ACTA Press, September 2004.

[13] A. M. McIvor and R. J. Valkenburg. Substripe localisation for improved structured light system performance, 1997.

[14] J. L. Posdamer and M. D. Altschuler. Surface measurement by space-encoded projected beam systems. *j-CGIP*, 18(1):1–17, January 1982.

[15] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno. A low cost 3D scanner based on structured light. In A. Chalmers and T.-M. Rhyne, editors, *EG 2001 Proceedings*, volume 20(3), pages 299–308. Blackwell Publishing, 2001.

[16] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 438–446, New York, NY, USA, 2002. ACM Press.

[17] K. Sato and S. Inokuchi. Three-dimensional surface measurement by space encoding range imaging. *Journal of Robotic Systems*, 2:27–39, 1985.

[18] L. Zhang, B. Curless, and S. M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *The 1st IEEE International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 24–36, June 2002.