FAST SUBPIXEL ACCURATE RECONSTRUCTION USING COLOR STRUCTURED LIGHT

Hao Li
Institut für Betriebs-
und Dialogsysteme
Universität Karlsruhe (TH)
Karlsruhe, Germany
email: hao@ira.uka.de

Raphael Straub
Institut für Betriebs-
und Dialogsysteme
Universität Karlsruhe (TH)
Karlsruhe, Germany
email: raphael@ira.uka.de

Hartmut Prautzsch
Institut für Betriebs-
und Dialogsysteme
Universität Karlsruhe (TH)
Karlsruhe, Germany
email: prau@ira.uka.de

ABSTRACT

Extracting range data through active optical triangulation requires robust stripe edge detection of the emitted pattern because of undesired shape and shading variations in the scanned object. We propose several extensions to a structured light system originally proposed by Zhang et al. [1]. Assuming stripes of a certain width, our edge detection criterion is achieved simply by restricting the number of consecutive edge pixels. Subpixel accuracy from one single input image can be obtained by approximating the gradient of the square local contrast with a piecewise linear function. Finally, we present a noise reduction technique for the range map through meshing followed by a parameterized face orientation culling. Experimental results have demonstrated major improvements in terms of robustness against acquisition noise, shading variations and complex shapes.

KEY WORDS

Active Vision, 3D Scanning, Reverse Engineering, Range Acquisition, Structured Light System, Optical Triangulation.

1 Introduction

Being able to reconstruct the shape of an object from a set of images has a tremendous impact on the field of computer graphics. Although a large assortment of commercially available 3D scanning devices can be employed for our system, we limit ourselves to devices which are relatively economical. We used an ASK M2 DLP video projector for the emitter and a Fuji Finepix S2 Pro professional digital camera for the sensor as shown in Figure 1. The projector produces a specific structured light pattern, in this case a set of stripes, which is then projected onto the target object. The digital camera ”sees” the structured light distorted if the object is not plane. Using these images combined with the relative positions and orientations of the sensor-emitter pair, we are able to extract the depth information using optical triangulation.

Some works involving structured light acquisition are described in [2, 3], but they usually require more input images. Moreover, if the sampled ranges reach a certain density, zooming in on the object will begin to cause aliasing effects, which is manifested as steps. This is due to the pixel-accurate edge detection in the sensored image which leads to a pixel-accurate triangulation. Typically subpixel-accurate edge detection requires multiple input images. A remarkable solution, which uses a sequence of a shifting illumination has been proposed by Curless and Levoy [4], namely the space-time analysis approach. However, using dynamic programming to compute correspondence can still cause errors such as mislabeled edges as a result of undetected edges and inappropriate shape of the object.

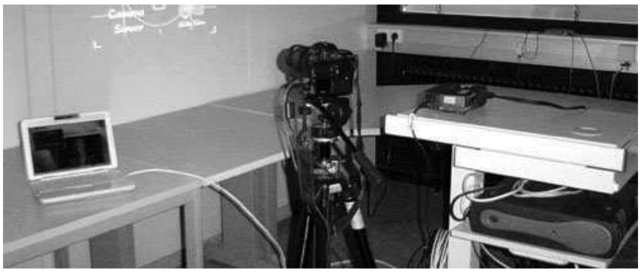


Figure 1. The optical triangulation devices.

To solve these problems, we propose a few modifications. More specifically, we are seeking to design a structured light system that is able to reach subpixel accuracy using one single input image, increase the robustness against mislabeling and achieve better edge detection. This paper is organized as follows. In section 2, we formulate the complete architecture of our acquisition system in a pipeline. In section 3, we briefly show some results from our color encoded triangulation system and discuss some remaining problems. Finally, a summary and propositions for future work are suggested in section 4.

2 A Shape Reconstruction Architecture

This paper focuses on range acquisition using a video projector and a digital camera. We obtain subpixel accurate range maps with a calibrated emitter-sensor pair. We need the camera and video projector extrinsic parameters (orientations and positions in space), as well as the intrinsic parameters, such as the focal length, before any triangulation can be performed. A notable solution to this non-trivial calibration problem has been proposed by Zhang [5], which can be adapted to a simultaneous self-calibration of the camera and the projector using stereo calibration techniques from [6] . Range acquisition is followed by a registration process which merges the different range maps in one point-cloud. Finally, we construct a mesh from the point-cloud and extract view independent textures in order to visualize material properties. An overview of the complete process is illustrated in Figure 2.

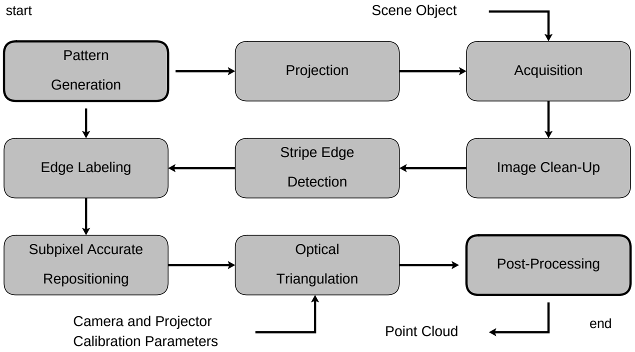
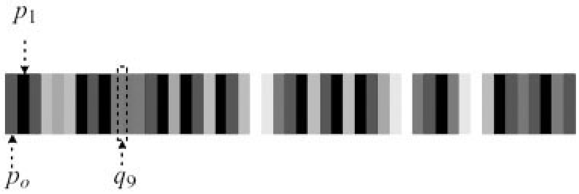


Figure 2. An overview of the shape reconstruction architecture.

2.1 Color Encoded Structured Light Pattern

To speed up the acquisition process, we project multiple stripes onto the object. To reliably identify the stripes in an image, it is necessary to carefully design an appropriate stripe pattern. We briefly review the color encoded stripe pattern based on De Bruijn sequences suggested by [1].



Let $\mathbf{p}_i = (p_i^r, p_i^g, p_i^b) \in \{0,1\}^3$ denote one of 8 different colors with p_i^c being its intensity in each channel $c \in \{r,g,b\}$. Then

$$\mathbf{P} = (\mathbf{p}_0, \mathbf{p}_1, \dots, \mathbf{p}_N)$$

represents a sequence of $N+1$ colored stripes projected onto the object. The first color $\mathbf{p}_0$ and the differences $\mathbf{d}_i = \mathbf{p}_{i+1} - \mathbf{p}_i$, computed in $\mathbb{Z}_2^3$, also represent $\mathbf{P}$ since $\mathbf{p}_{i+1} = \mathbf{p}_i + \mathbf{d}_i = \mathbf{p}_i \,\mathsf{XOR}\, \mathbf{d}_i$. Because successive stripes have different colors, we require $\mathbf{d}_i \neq [0,0,0]^t$. To enhance recognition, we choose $\mathbf{P}$ such that $\mathbf{D} = (\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{N-1})$ is a k-ary De Bruijn sequence of order n, where $k = 7$ and $N < k^n$. In a k-ary De Bruijn sequence of order n, any sequence of n or more consecutive elements appears only once. Hence the maximum length is k^n. We precompute $\mathbf{D}$ for different values of n with the recursive algorithm available online [7].

2.2 Input Image Clean-Up

Detecting edges between stripes is hampered by noisy images due to sampling quantization, unfavorable lighting circumstances, object textures, camera CCD capturing noise and especially the low pixel density of the projector's LCD. Also, it is only possible to project a sharp image at a specific distance which is dictated by the depth of focus of the projector. Another problem is that the shape of the object might be self-occluded and the color of the projected pattern might not exactly correspond to the acquired one. All these factors make consistent edge detection difficult. Hence, we propose to purge the images in three steps.

First, we take care of the color crosstalk phenomenon, which is often due to uncalibrated colors projected by the emitter and acquired by the sensor and is also due to the surface that modifies the projector color spectrum in unknown ways. As proposed by Caspi et al. [3], we assume that the reflected light depends linearly on the emitted light in each color channel and that the instruction color matches perfectly the emitted color. Hence, we can normalize the image color separately in each pixel by linear transformation such that the black light and the white light projections have black and white images respectively. Since the ambient light, which is generated by a black light projection, is neglectible, we assume that black light emission leads to black images with all color values equal to zero. Furthermore, the acquisition process is performed in a completely dark room. This reduces the intervention from light sources other than the projector light.

Second, we remove dark areas from the image, where the light intensity is below a user defined threshold. This eliminates the background and pixels hidden to the emitter.

Third, an optional final smoothing can be applied if the acquired image remains too noisy for further processing. Assuming an additive white Gaussian noise over the image, we filter out high frequency noise using a simple approximated low-pass filter. This noise reduction pass should be used carefully as excessive smoothing can make the stripe edge detection even more difficult.

2.3 Color Stripe Edge Detection

2.3.1 Multi-spectral Definition of Edges

For edge detection to be as consistent and as accurate as possible, it is crucial to have a formal definition of edges in multi-spectral images. We use Cumani's definition of an edge [8] in color images and we assume that the vertical oriented stripes w. r. t. the camera form the edges we are looking for. This implies that only a 1-dimensional edge

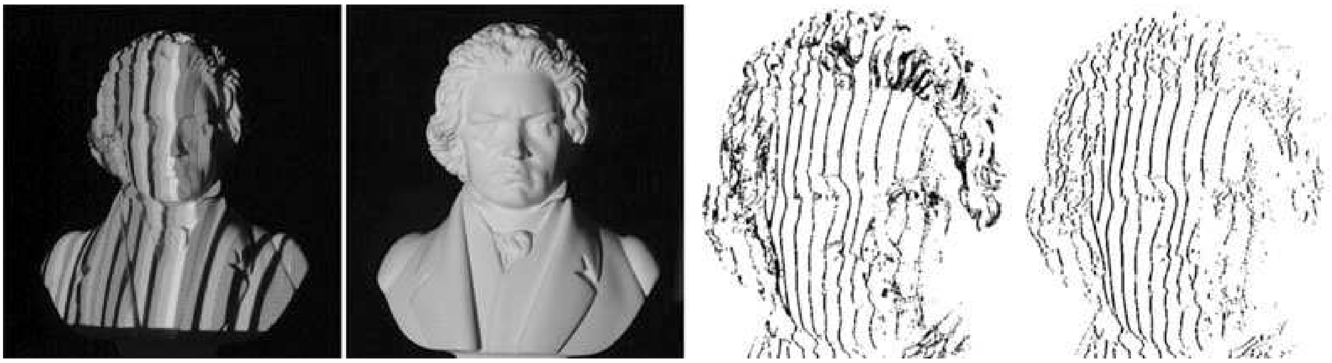


Figure 3. Input image of a structured light projected Beethoven bust (a). A shot of the white pattern projection (b) is required for the colorimetric calibration. A comparison of the edge-detection is shown in the next two figures. A visualization of the color edge detection (c). An improvement can be attained by restricting 5 consecutive detected edge pixels (d).

detection in the rows of the image is required. Let $\mathbf{r}_i$ be the rgb-color vector of the ith pixel. The square local contrast $c_i = (\Delta\mathbf{r}_i)^2 = \|\mathbf{r}_{i+1} - \mathbf{r}_i\|_2^2$ corresponds to an edge if $c_i - c_{i-1} > 0 > c_{i+1} - c_i$. This implies that edges are separated by at least two pixels.

2.3.2 Consecutive Edge Pixels Restriction

Edge detection as described in the previous section might not suffice for our application. We might detect undesirable shading contrasts in addition to color transition contrasts. Therefore, we report an edge only if no other edge with higher square local contrast is found within the next M pixels, where $M \geq 2$. We set M to be slightly smaller than the average stripe width in the image so that the advantage of detecting the correct edges outweighs the drawbacks of not detecting an edge as demonstrated in Figure 3 with $M = 5$.

2.4 Edge Labeling

With the labeling process, we re-identify the projected edges from the observed image. The method based on multi-pass dynamic programming proposed by [1] is very well suited for solving the problem of labeling multiple color-encoded edges, especially in the case of holes and self-occlusions in the scanned surface that yield undetectable edges.

Labeling means to match for each scan-line the projected stripe sequence with the captured stripe sequence. We match the stripe color transitions and obtain the projected color transition sequence

$$\mathbf{q}_j = \mathbf{p}_{j+1} - \mathbf{p}_j = \begin{bmatrix} q_j^r \\ q_j^g \\ q_j^b \end{bmatrix} \in \{-1,0,1\}^3, \quad j = 0,\ldots,N-1,$$

and similarly the captured color transition sequence

$$\mathbf{e}_i = \begin{bmatrix} e_i^r \\ e_i^g \\ e_i^b \end{bmatrix} \in [-1,1]^3, \quad i = 0,\ldots,M-1 \quad .$$

A labeling is defined by a set of P matches

$$\Phi = \{(j_1,i_1),...,(j_k,i_k),...,(j_P,i_P)\}$$

with

$$(j_k,i_k) \in \{0,\ldots,N-1\} \times \{0,\ldots,M-1\}$$

and

$$j_1 < j_2 < \ldots < j_P \quad , \quad i_1 < i_2 < \ldots < i_P \quad .$$

Consistency between a detected edge $\mathbf{e}_i$ and projected edge $\mathbf{q}_j$ is measured by a scoring function defined as

$$\text{score}(\mathbf{q}_j,\mathbf{e}_i) = \min_{c\in\{r,g,b\}}\{\text{consistency}(q_j^c,e_i^c)\}$$

with consistency(q_j^c,e_i^c) as illustrated in Figure 4. Thus, the score of a labeling is defined as

$$\sigma(\Phi) = \sum_{k=1}^{P} \text{score}(\mathbf{q}_{j_k},\mathbf{e}_{i_k})$$

and the optimal labeling is given by

$$\Phi^* = \arg\max_{\Phi}\{\sigma(\Phi)\} \quad .$$

Typically for dynamic programming approaches, we recursively compute a cost matrix $S = [s_{ji}]$ containing the scores of optimal sub-labelings as follows:

$$s_{ji} = \begin{cases} 0 & \text{, if } j = 0 \text{ or } i = 0 \\ \max\{s_{j-1,i-1} + \text{score}(\mathbf{q}_j,\mathbf{e}_i),\, s_{j-1,i},\, s_{j,i-1}\} \end{cases} \quad .$$

Finally, we backtrack the cost matrix to obtain an optimal labeling.

In practice, we often observe mislabeled edges. We must expect repeating subpatterns of size smaller than n within our De Bruijn pattern of order n. Also in many cases, the scanned object covers only part of the complete stripe pattern. Moreover, our dynamic programming approach evaluates the cost matrix in a bottom up fashion, which corresponds to a scanning direction from left to right. These situations often cause wrong matches. The problem lies in the lack of stripe information caused by the empty background of the shape and by shadows. Putting a white plane behind the object during the stripe projection and caption therefore reduces the effect of the empty background. In this way, the previously missing stripes can be recovered on the wall and will fix the ambiguity problem.
To solve the occlusion problem, we use the multi-pass procedure proposed in [1]. This increases the runtime by a factor of order O(N). In practice, objects that contain too many occlusions might already fail the edge detection stage due to high shading frequencies that increase with the object's shape complexity.

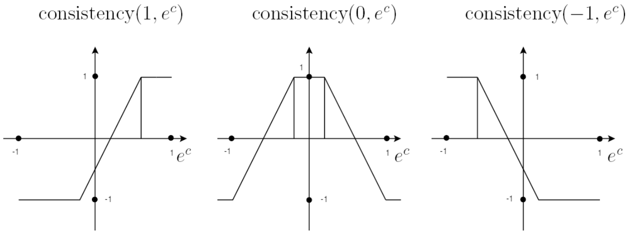


Figure 4. The consistency function.

2.5 Subpixel Accurate Repositioning

For the optical triangulation pass, we use our previously labeled edges that were detected in the input images. These input images are pixel arrays recovered from the CCD of the camera. It is obvious that from a pixel accurate edge detection, we are only able to triangulate the image at pixel accuracy. This imprecision is manifested by jagged aliasing effects when the final object is zoomed in. In order to produce high precision scanned results, many approaches have been suggested. Zhang et al. [1] adapt the space time analysis technique on this colored structured light approach with the disadvantage of requiring multiple input images of a shifted De Bruijn pattern. We show how edges with subpixel accuracy are detected from a single image. This will consequently produce a subpixel accurate range map. An edge that has been detected between two pixels typically is positioned exactly between these two pixels. To avoid the undesired quantization errors, we exploit the light intensities of these pixels to approximate a subpixel accurate position, similar to the linear boundary interpolation approach described in [9]. As explained in Section 2.3.1, we detect an edge if $c_i - c_{i-1} > 0 > c_{i+1} - c_i$. Rather than positioning the edge at $x = i + \frac{1}{2}$, we place it at the subpixel position

$$x = \frac{(i+1)\cdot(c_i - c_{i-1}) - i\cdot(c_{i+1} - c_i)}{2c_i - c_{i+1} - c_{i-1}} \quad .$$

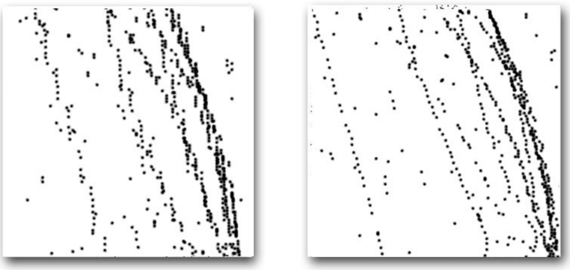


Figure 5. Pixel accurate optical triangulation without (left) and with subpixel accurate repositioning (right).

2.6 Optical Triangulation

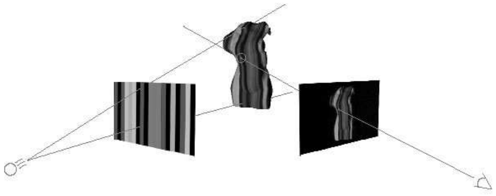


Figure 6. A projected stripe transition represents a plane and a pixel on the CCD of the camera represents a ray of sight. Depth is extracted by computing a ray-plane intersection.

Once we successfully label the detected edges on the captured image with the corresponding stripe transitions on the projected pattern, we are able to compute the depth information of these edges. The vertical stripe transition pattern we are projecting on the object represents a set of vertical planes passing through the central point of the projector. On the other hand, an edge on the sensor image corresponds to the line of sight that intersects the projected planes in one single point. The optical triangulation with a stripe pattern has been reduced to a ray plane intersection, which is straightforward to compute.

2.7 Artifact Elimination through Meshing

In our experiments, we observed that most areas of the range map are still seriously affected by noise. Badly triangulated points are scattered all over the object. This is due to the difficulty in detecting the projected stripe edges

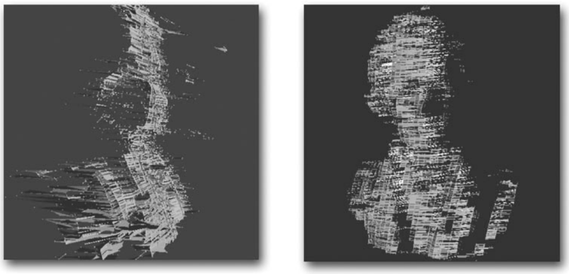


Figure 7. After meshing the triangulated points, artifacts appear in the visualization of the triangle surface (left). An additional orientational face culling reduces most artifacts due to wrongly labeled or detected edges (right).

in high frequency shading areas, that results in bad labeling and subsequent wrong triangulation. We therefore designed a post-processing algorithm to annihilate the triangulated points that are unlikely to be part of the object. Although this might cause fewer triangulated points, i. e., more holes in the end result, it is more important to acquire correct than more data. If the *i*th projected edge is detected in row j of the image, we compute a point $\mathbf{p}_{ij}$ of the object in space as described in section 2.6. Two points $\mathbf{p}_{ij}$ and $\mathbf{p}_{kl}$ are called neighbors if $(k,l)-(i,j) \in \{-1,0,1\}^2$. Three points that are neighbors to each other form a triangle. First, we remove those triangles that are not visible to the camera and the projector. This is simply a back-face culling algorithm that computes the scalar product of the triangle surface normal with the normalized direction of the line of sight to the center of this triangle. Triangles with negative values are omitted. Furthermore, we leave out back-faced surfaces and surfaces with normals that are almost orthogonal to the direction of the line of sight. In practice, depending on the shape of the object, an angle α of up to 60 degrees between the surface normal and the line of sight vector is accepted. Second, we remove all points $\mathbf{p}_{ij}$ that are not part of any triangle.

3 Results

Our experiments focus on the acquisition of the Beethoven bust shown in Figure 3 and a skull model (Figure 9). The complete acquisition process takes less than a minute of computation time.

Results have shown a remarkable improvement in terms of subpixel accuracy. While pixel accurate triangulated point clouds show a jagged aliasing effect when zoomed in, an approximated gradient of the square local contrast produces smooth transitions within neighboring ranges as illustrated in Figure 5.

We also observed that the edge detection, based on a restricted consecutive edge pixel occurrence is quite effective when it comes to annihilating artifacts in high frequency shading areas (Figure 3). The only disadvantages are the additional parameter that has to be set manually and the few less detected edges. This step is indispensable for a usable stripe edge extraction from complex shape objects.

The post-processing through meshing and face orientation culling also has a radical impact in terms of noise reduction. A large amount of wrongly triangulated ranges can be removed (Figure 8), but also at the cost of an additional parameter and less extracted ranges. Multi-pass dynamic programming, which solves the matching problem, has achieved satisfactory results except at the boundary of the objects. Due to the greedy property of the backtracking phase in dynamic programming, undesired matches are added for a global optimal solution. These artifacts are either removed by post-processing meshing as mentioned previously or by setting a plane wall behind the object to recover the unreflected stripes on the object. This simple solution enables the dynamic programming algorithm to start and end with correct matches as demonstrated in Figure 8. However, this solution has the drawback of requiring an additional frame without the background to extract its silhouette. This frame is taken from the white illumination of the colorimetric calibration. To circumvent this additional frame, we can use a depth threshold w. r. t. the camera's position. Every point above a certain threshold is omitted.

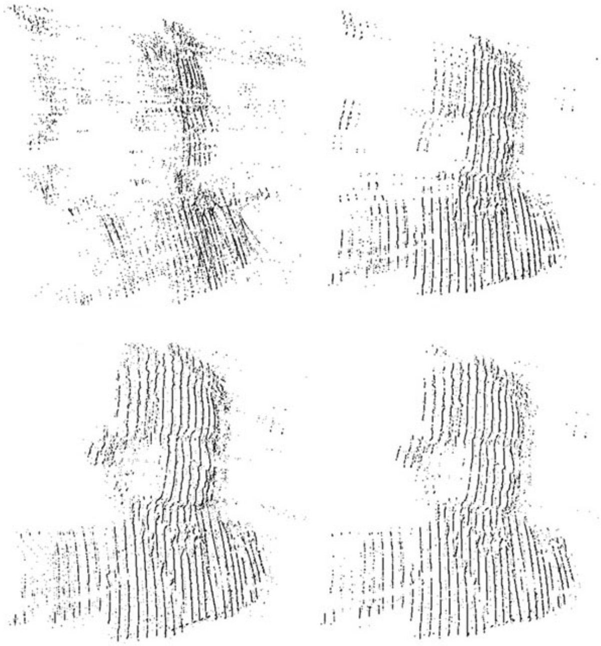


Figure 8. Point cloud without (top) and with (bottom) white background plane and artifact elimination for $\alpha = 90^\circ$ (left) and $\alpha = 60^\circ$ (right).

While improvements have been achieved with a few extensions, it is still difficult to detect correct edges when it comes to stripe segmentation from the capture. This is due to the high curvature areas in the object and unsuitable material reflection properties. Moreover the projector is only able to project sharp images from a specific distance to the object. In other words, only edges on a plane at a specific distance are sharp, which is restricted by the depth of focus.

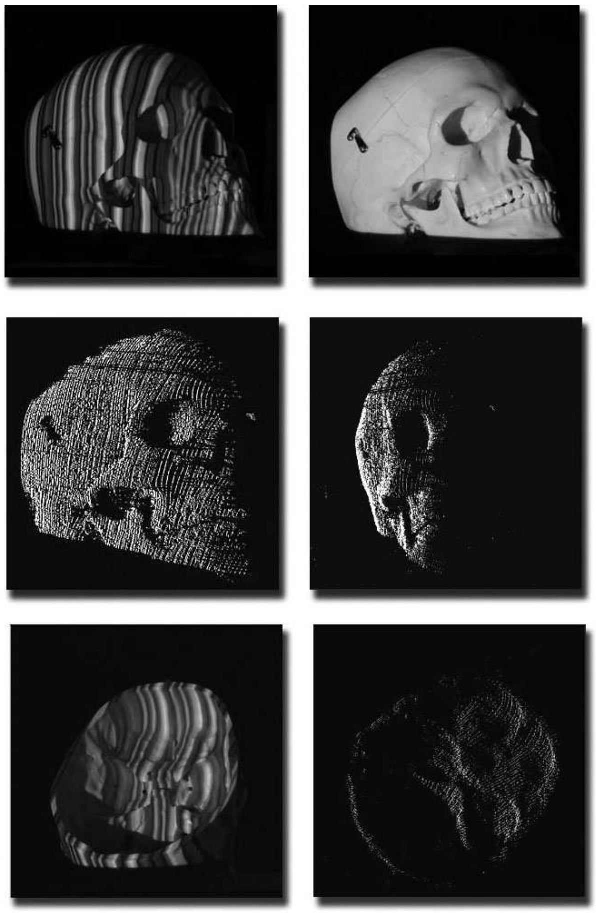


Figure 9. Reconstruction of a skull model (top) and its basis cranii (bottom) using a pattern of 125 stripes. Each point cloud is obtained from only two input images.

4 Conclusion and Future Work

In this work, we have examined the structured light system proposed by Zhang et al. [1], which uses multi-pass dynamic programming to solve the correspondence problem, and we have performed some modifications to achieve subpixel accuracy, less erroneous matching during the labeling and better robustness against noise. Subpixel accuracy from one single frame is attained by approximating the gradient of the square local contrast with a piecewise linear curve. The labeling process is stabilized at the object's boundary by considering a plane background during the pattern projection acquisition. The noise is reduced with an extended edge detector and a post-processing procedure based on meshing and face orientation culling.

With the aim of less human inputs, we intend to integrate a simultaneous self-calibration for the projector and for the camera, including an off-line colorimetric calibration for real one-shot pattern acquisition. In addition, we hope to find a way to automatically determine the required thresholds during the edge detection by analyzing statistical properties of the color and light intensities of the images. Finally, we hope to complete the structured light scanning system with multi-view registration and texture reconstruction stages, both equally important.

5 Acknowledgments

Special thanks go to Yuanshan Lee for reviewing the paper.

References

[1] 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*, pp. 24–36, June 2002.

[2] C. Rocchini, P. Cignoni, C. Montani, P. Pingi, and R. Scopigno, "A low cost 3D scanner based on structured light," in *EG 2001 Proceedings* (A. Chalmers and T.-M. Rhyne, eds.), vol. 20(3), pp. 299–308, Blackwell Publishing, 2001.

[3] D. Caspi, N. Kiryati, and J. Shamir, "Range imaging with adaptive color structured light," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 470–480, 1998.

[4] B. Curless and M. Levoy, "Better optical triangulation through spacetime analysis," in *ICCV*, pp. 987–994, 1995.

[5] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[6] D. Demirdjian, A. Zisserman, and R. Horaud, "Stereo autocalibration from one plane," in *ECCV (2)*, pp. 625–639, 2000.

[7] F. Ruskey, "The combinatorial object server. http://www.theory.csc.uvic.ca/cos/," 2000.

[8] A. Cumani, "Edge detection in multispectral images," *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, vol. 53, pp. 40–51, January 1991.

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