Real-time surface tracking with uncoded structured light

Willie Brink

Council for Scientific and Industrial Research, South Africa
wbrink@csir.co.za

Abstract

A technique for tracking the orientation and position of a near-planar surface in real-time is presented. It uses the principle of structured light to determine 3D surface shape, by the projection of a dense pattern of uniform (uncoded) stripes. In order to estimate pose a weighted least-squares plane is fitted to the 3D data. As an example the technique is applied in a gaming environment where the user moves and rotates his/her open hand in the field of view of the structured light system to control the yaw, pitch and speed of a model aircraft in real-time.

1. Introduction

High-speed 3D shape measurement and tracking has immense potential in a wide variety of application fields [1], ranging from medical imaging and industrial testing to mobile robot navigation and the gaming industry. Such a system may involve two steps: (1) reconstructing or estimating the shape of some dynamic target surface, and (2) approximating and tracking its orientation and position over time.

The first step is usually regarded as a computer vision problem, and addressed through the utilization of a 3D imaging method such as stereo vision [1, 2] or structured light [3, 4]. The latter has the advantage over the former that its output resolution and accuracy can be controlled to a far greater extent. On the other hand, unlike stereo systems that require only two cameras, structured light is an active vision system that has to project a pattern onto a target surface, that has to be opaque and non-specular, in order to measure it. If the specific application permits, however, structured light is ideal for high-speed, low-cost and high-quality 3D surface acquisition.

A structured light system consists typically of a projector (for casting some pattern onto the target surface) and a camera (for capturing the reflecting pattern). The pattern is found in the image and its deformation reveals the shape of the target. The concept evolved from single laser-spot projection, requiring several hours to complete a scan, to the projection of complex patterns that can measure large surface areas in a few milliseconds. For real-time systems it is imperative that sufficient information for full surface reconstruction is gathered within the timebase of a single video frame. A dense pattern of parallel stripes is often chosen as a projection pattern because it can cover a large surface area in a single shot and can produce high-quality 3D models.

The problem associated with multiple stripe patterns, here called the *indexing problem*, amounts to establishing correct correspondences between the projected stripes and those observed in the recorded image. Various different approaches have been proposed, including coding stripes by colour [5], width [6] and time-dependent sequences [4]. These have limitations however: colour cannot be applied consistently to surfaces with weak or ambiguous reflectance, for width coding the resolution is less than for uniform narrow stripes, and time-coded sequences require multiple images over time and are thus not suitable for real-time applications. In light of this we are biased to the use of *uncoded* (i.e. homogeneous) stripes. The indexing problem becomes more involved but can still be solved with a high degree of accuracy [7].

Once a 3D model is constructed the target surface can be tracked over time as it moves, rotates and deforms. We accomplish this by fitting a weighted least-squares plane to the data and tracking its orientation and position in 3D space. As an example for application the technique is implemented in a human-computer interaction (HCI) system where a user controls the yaw, pitch and speed of a model aircraft in real-time, by moving and turning his/her open hand in the field of view of the structured light system.

The remainder of the paper is structured as follows. Section 2 discusses the structured light system we use for 3D reconstruction, a calibration technique and how a surface model can be built. Section 3 explains the method for pose estimation, provides some simplifications in order to speed up the system for real-time applications, and describes the HCI system developed for controlling a model aircraft. The paper is concluded in section 4.

2. Structured light

A schematic diagram of a typical structured light system is shown in Fig. 1. A light stripe is projected onto some target surface, and captured by a camera. The stripe is extracted from the image and, together with the spatial relationship between the projector and camera, reveals the 3D shape of the surface along that stripe. This system uses a single stripe to measure one "slice" at a time, and the target object can now be moved or rotated repeatedly in order to measure other parts that can finally be stitched together into a surface model.

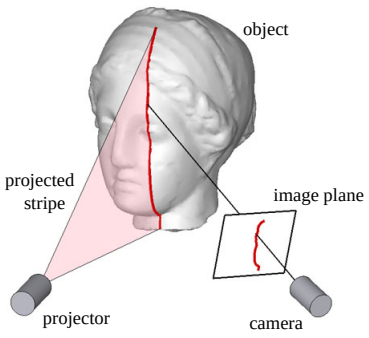


Figure 1: *Schematic diagram of a structured light system where a light source projects a stripe onto a target object's surface and a camera captures the scene.*

The above procedure can be very effective in generating accurate and high-resolution surface models of stationary objects. It is slow however, and does not lend itself to real-time applications where complete surface information has to be acquired at frame rate (i.e. within a few milliseconds).

The current availability of controlled light sources, such as DLP projectors, allows for more complex patterns that increase the surface area measurable per single scan. A dense pattern of parallel stripes is a popular choice [8], but its reconstruction accuracy depends upon establishing correct correspondence between projected and recorded stripes (so that, as shown in Fig. 1, surface points can be determined as intersections with the correct stripe planes). Different methods have been devised to discriminate between captured stripes by, for example, varying colour or width. However we are interested in patterns of homogeneous stripes as they can be applied consistently on surfaces with weak or ambiguous reflection and produce maximal resolution [7].

2.1. Calibration

Calibration of the structured light system is necessary in order to extract accurate metric information. Without calibration a surface can be reconstructed only up to some projective transformation [9] which does not preserve distance.

The calibration procedure should produce the intrinsic parameters (such as the focal lengths, principal points and distortion coefficients) of the camera and projector, as well as the relative pose between the camera and projector (often referred to as the extrinsic parameters). The parameters can be determined once prior to surface acquisition under the assumption that they remain fixed throughout operation.

Figure 2 depicts the layout of the structured light system and our defined coordinate system. The camera is located at the system origin and the projector at point $\mathbf{p}$. The camera ray associated with pixel (r, c) in the image will be denoted by $\mathbf{r}(r, c)$, and the projected plane that produces stripe n will have normal vector $\mathbf{t}(n)$.

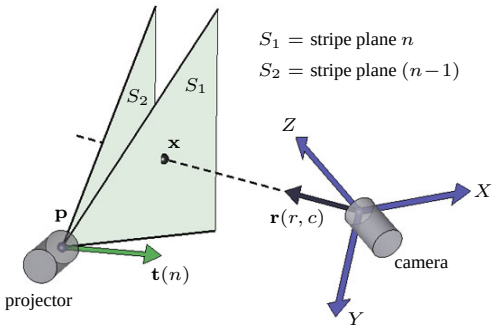


Figure 2: *We let the system coordinate frame (X, Y, Z) coincide with the camera's. The projector is located at point $\mathbf{p}$ and stripe plane n has normal $\mathbf{t}(n)$. The camera ray associated with pixel (r, c) is denoted by $\mathbf{r}(r, c)$.*

The camera's intrinsic parameters can be found through Zhang's method [10] (for example). A printed grid pattern is attached to a planar surface and captured under different orientations. Figure 3a shows one such image. The grid is extracted from the images by a corner detector (e.g. [11]) and the camera's parameters are then estimated using a closed-form solution. The parameters can be refined further through the minimization of a non-linear functional. We let the camera's principal point coincide with the system origin. After calibration any pixel (r, c), i.e. at row r and column c, in a captured image can be associated with a ray $\mathbf{r}(r, c) \in \mathbb{R}^3$. The surface point $\mathbf{x}$ captured at (r, c) is then located at $\lambda\,\mathbf{r}(r, c)$, as can be seen in Fig. 2, for some unknown $\lambda > 0$.

The projector may be calibrated as follows. A grid pattern is projected onto a planar surface within the camera's field of view and an image is recorded. Such an image is shown in Fig. 3b. The location and orientation, in system coordinates, of the planar surface is determined by means of physical markers at known distances apart (the four outer corner markers in Fig. 3b). The locations of the projected grid points are found by computing intersections of this plane with the camera rays corresponding to their pixel coordinates. Repeating the process for different orientations of the plane yields a collection of points in space that can be used to trace back the position $\mathbf{p}$ of the projector. A row (or column) of the projected grid can then be used to determine the relative orientation of the projector. Here it is assumed that the projector counters the effects of radial distortion internally, which is true for most DLP projectors and can be verified easily by observing that the projection of any straight line onto a planar surface remains straight.

The stripe pattern is typically an image consisting of alternating black and white bands of pixel rows, displayed at full-screen resolution. A single stripe can be associated with some index n, directly related to a row (or column) in the pattern, and the calibration parameters can be used to determine a normal vector $\mathbf{t}(n)$ of the plane containing that stripe (see Fig. 2).

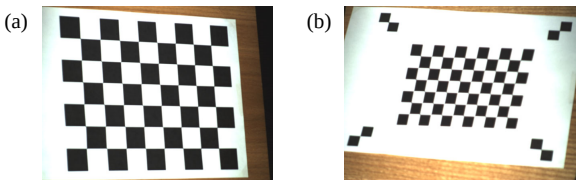


Figure 3: *Examples of captured images for calibrating (a) the camera and (b) the projector. In (b) the four outer corners are physically printed on the plane, and the inner grid is projected by the projector.*

2.2. Indexing uncoded stripes

Before 3D surface points can be determined (as Fig. 2 illustrates, by intersections of camera rays with projection planes) the stripes in the recorded image need to be (1) located and (2) indexed. The latter refers to the assignment of indices to located stripe pixels, which correspond to the individual projected stripes. For example, we may choose to assign an index 0 to the centre stripe and increase (decrease) the index as the row in the projection pattern increases (decreases). An index n can then be mapped to its corresponding stripe plane.

To locate the stripes in the image a simple and quick linear search for local maxima in luminance values across every column can be sufficient, where additional local thresholding can be applied to reduce the effects of noise.

The method we developed for indexing dense patterns of uncoded stripes is described in detail in [7] and a brief summary is given here. The method relies on some geometric constraints derived from the epipolar geometry [12] of the system. Located stripe pixels are grouped according to "left–right" adjacencies

(meant here in an 8-connected sense). Certain smoothness assumptions on the surface are made which then permit the assignment of a single index to all pixels in a connected group. "Up–down" adjacencies between the groups are evaluated in order to build a weighted directed graph. Here edge weights favour clear (and likely to be correct) index transitions between groups. A traversal of a maximum spanning tree of the graph yields an indexing of all the groups relative to an arbitrary starting point. These relative indices differ from their true values by some constant K found, in our implementation, through locating a single reference stripe (noticeable in the image as being slightly darker than the rest) with known index.

The abovementioned smoothness assumption regards elements of an apparently continuous stripe in the image as the same projected stripe. Discontinuities in surface depth of a certain critical size can nullify this assumption, introduce indexing errors and necessitate stripe coding [7]. But we aim to reconstruct near-planar surfaces that should not contain such discontinuities and, because the indexing method seeks a solution that agrees with stripe connections and transitions correctly on a global scale, errors tend to be small, localized and have little influence on the pose of the reconstructed model.

Figure 4 shows an example image of a statue's head, and a close-up in which the dense stripe pattern can be seen. The located stripe pixels are shown on the right, coloured according to assigned indices (note that the colour sequence repeats).

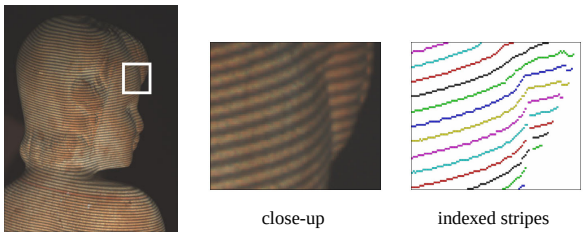


Figure 4: *A captured image and a close-up, in which the dense stripe pattern can be seen. The stripes need to be located and indexed, as shown on the right, for reconstruction.*

2.3. 3D reconstruction

The above procedure yields a collection of indexed stripe pixels, each having pixel coordinates (r_i, c_i) and an index n_i. We determine each one's associated surface point $\mathbf{x}_i$ as the intersection of the camera ray $\mathbf{r}(r_i, c_i)$ with stripe plane n_i having normal $\mathbf{t}(n_i)$. Hence

$$\mathbf{x}_i = \left(\frac{\mathbf{t}(n_i) \cdot \mathbf{p}}{\mathbf{t}(n_i) \cdot \mathbf{r}(r_i, c_i)} \right) \mathbf{r}(r_i, c_i), \qquad (1)$$

where $\mathbf{p}$ indicates the position of the projector (see Fig. 2).

3D points can be calculated for all indexed stripe pixels and a piecewise linear surface can then be constructed in the following way. Suppose the recorded image has N columns, let P indicate the total number of indexed stripes and $n_{\min}$ the smallest index found. Let $\mathbf{x}(i, j)$ indicate the surface point on stripe $n_i = i + n_{\min} - 1$ captured on image column j (we know from the epipolar constraints that an index can exist at most once in each image column). If such a point does not exist, i.e. if stripe n_i is not present in column j, $\mathbf{x}(i, j)$ is flagged as "missing". $\mathcal{S}$ is then defined to be the quadrilateral mesh consisting of all 4-sided polygons with vertices $\mathbf{x}(i, j)$, $\mathbf{x}(i, j+1)$, $\mathbf{x}(i+1, j+1)$ and $\mathbf{x}(i+1, j)$, with $i = 1, \ldots, P-1$ and $j = 1, \ldots, N-1$. Quadrilaterals containing missing points as vertices are removed from $\mathcal{S}$.

An example of a 3D surface reconstructed by this method is shown in Fig. 5, from the source image shown on the left. The model is shown from different viewpoints and consists of roughly 100,000 vertices.

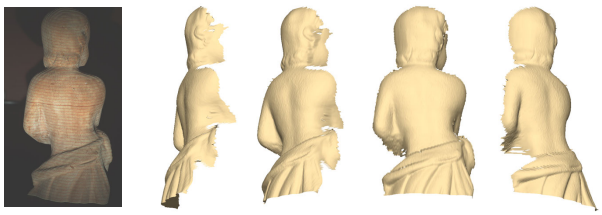


Figure 5: *The surface reconstructed from the single image on the left, shown here from 4 different viewpoints.*

Note that the resolution (or smallest measurable interval) of the system depends on the resolution of the image, as stripes are found only to pixel accuracy. A so-called sub-pixel estimator can be applied [13] which has been shown to greatly improve the resolution. Further surface processing such as hole-filling, smoothing, subdivision, etc., can also be performed to enhance the visual appearance of the 3D model.

3. Real-time surface tracking

The aim of this work is to track the position and orientation, which we collectively refer to as *pose*, of a near-planar surface such as an open hand. Figure 6 shows an image captured of a hand on the left and its reconstruction on the right.

3.1. Pose estimation

We opt to approximate the pose of the hand by a weighted least-squares plane. The palm should influence the orientation of such a plane more than the fingertips, hence weights are assigned according to the distance from the mean of the collection of points (assumed to be close to the centre of the palm). Suppose $\mathbf{x}_i = (x_i, y_i, z_i)$, $i = 1, \ldots, n$ are the 3D surface points, and let $\mathbf{m}$ denote their mean. The Euclidean distance between $\mathbf{x}_i$ and $\mathbf{m}$ is denoted by d_i. We scale these distances to the interval $(0, 1)$ and subtract from 1 to arrive at weights

$$w_i = 1 - \frac{d_i - d_{\min}}{\max_j\{d_j - d_{\min}\}}, \quad \text{with } d_{\min} = \min_j\{d_j\}. \qquad (2)$$

The equation of a plane is $ax + by + cz + d = 0$ which, assuming that $c \neq 0$, may be rewritten as $\alpha x + \beta y + \gamma = z$. The unknowns α, β and γ can be found through a least-squares solution of the overdetermined system

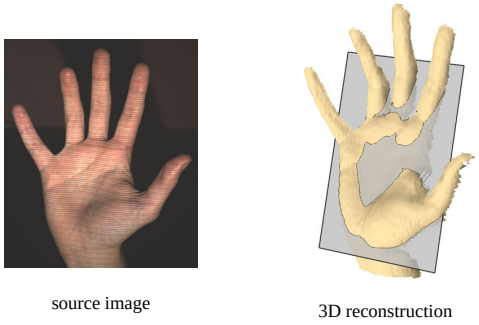
$$W \begin{bmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = W \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}, \qquad (3)$$

with W the $n \times n$ diagonal matrix containing w_i, $i = 1, \ldots, n$ on the diagonal. Note that the contribution of each point is actually weighed by w_i^2 in the sum minimized by the least-squares solution to the above system.

The plane can now be defined completely by its normal vector $\mathbf{n}$ and a point $\mathbf{q}$ on it, where

$$\mathbf{n} = \begin{bmatrix} \alpha \\ \beta \\ -1 \end{bmatrix} \quad \text{and} \quad \mathbf{q} = \begin{bmatrix} m_x \\ m_y \\ \alpha m_x + \beta m_y + \gamma \end{bmatrix}. \qquad (4)$$

Here m_x and m_y indicate the x- and y-coordinates of $\mathbf{m}$ respectively and, as it stands, $\mathbf{n}$ is not normalized. Note that the assumption that the z-component of $\mathbf{n}$ will never be zero is valid because the structured light system cannot reconstruct surfaces parallel to the Z-axis. An example of a weighted least-squares plane is shown in Fig. 6 (right).



source image 3D reconstruction

Figure 6: *The pose of the hand is approximated by a plane in a weighted least-squares sense.*

The plane estimates one translational and two rotational degrees of freedom (DOF) of the pose of the hand. We aim to extract this information and apply it to some computerized model in real-time. The structured light system used for the examples in Fig. 5 and Fig. 6 utilizes a colour camera with resolution 1024×768, and projects roughly 120 stripes across the surface of the hand (depending, of course, on the hand's distance from the projector). It yields a 3D model consisting on average of about $50,000$ points. The time required for the entire process by our implementation in C, executed on a Pentium 1.79 GHz dual-core processor with 2GB RAM, is given in Table 1 (the process of capturing an image involves sending a request to the camera, waiting and receiving camera output).

process	time (s)	Hz
capture image	0.018	56
locate stripes	0.053	19
index stripes	0.087	11
calculate 3D points	0.028	36
determine pose	0.019	53
TOTAL	0.205	5

Table 1: *The execution times required by the various processes in estimating the pose of an open hand.*

3.2. Simplifying the system for real-time

Although fast, particularly when considering the size of the generated models, 5 frames per second (fps) is hardly real-time. Since we are chiefly interested in the pose of the hand, and in an effort to speed up the process, the number of points in the output data can be reduced drastically. A simple test shows that a reduced version of the 3D model in Fig. 6, consisting of only about 2,500 points instead of 50,000, produces a least-squares plane very close to the one fitted to the original data. The normals differ by only about 0.01 degrees and the points determined through (4) are only 0.3*mm* apart.

Our simplified system captures 640×480 images, projects roughly 35 stripes across the surface of the hand and generates about $2,500$ points. Each image now contains almost 0.4 times as many pixels and the number of stripes that need to be located and indexed is about a third of the number in the original system. All the algorithms can be implemented in linear time complexity, except for solving (3) which is quadratic in n. The execution times achieved by the simplified system are given in Table 2.

process	time (s)	Hz
capture image	0.011	91
locate stripes	0.012	83
index stripes	0.016	63
calculate 3D points	0.006	167
determine pose	0.004	250
TOTAL	0.049	20

Table 2: *The execution times required for the simplified system to estimate the pose of an open hand. Experimental tests show that the result is insignificantly different from that of the original system.*

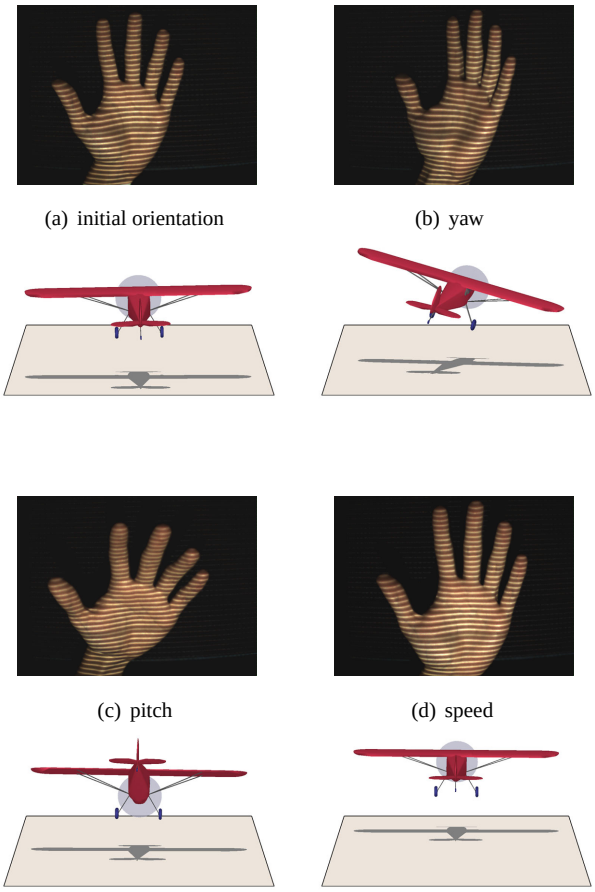
This technique for estimating the 3D pose of a hand runs at about 20 fps which can be sufficient for real-time systems. To demonstrate a possible application we applied the resulting poses to a model aircraft in a gaming environment; see Fig. 7. The camera and projector are positioned next to the computer screen and face towards the user. Note therefore that the hand in the images captured (and shown in the figure) are the left hand of the user, and the pictures of the model aircraft are shown as the user would view them on the screen.

A reference plane is defined for the aircraft which is parallel to the X–Y plane in system coordinates when the aircraft faces forward (away from the user, e.g. Fig. 7a). The differences in rotation and translation from the previous frame are determined, the plane is transformed at the current frame to coincide with the estimated pose of the hand, and the aircraft is rotated and moved accordingly. In this manner two rotational DOF can be controlled (Fig. 7b and c), and one translation DOF that may be used to control the forward speed (Fig. 7d).

The three DOF mentioned can be sufficient for this application, and provide the user with a immersive sense of control. The movements of the aircraft appear continuous and smooth, due to the system operating at approximately 20 fps (the time required to apply the calculated pose to the aircraft and render the result is short relative to those listed in Table 2).

The two remaining translational DOF (those that allow two-dimensional movement within the plane) could be incorporated as well by simply tracking the bounding box of the hand in the image. These could then be used to move the aircraft left, right, up and down. The remaining rotational DOF (allowing in-plane rotation) is slightly more difficult and would require, for example, some analysis of the actual 2D shape of the hand and how it rotates about the normal of the least-squares plane from one frame to the next.

Note that in the current system we do not attempt to first segment the hand from background regions of the image be-



(a) initial orientation (b) yaw

(c) pitch (d) speed

Figure 7: *The position and orientation of a hand is tracked in real-time and shown here to control the yaw, pitch and speed of a model aircraft. Note that the images of the hand are shown from the camera's point of view and those of the model aircraft from the user's.*

cause (it is assumed that) the background is far away. Stripes projected on background objects should therefore be out of focus so that they are not detected in the image and hence will not appear in the reconstructed model.

Also, stripes are currently located and indexed in every image frame completely independently. A further speed-up may be possibly if some temporal consistency constraint is incorporated, e.g. one that in some way bounds the spatial displacement of a particular stripe from one frame to the next.

4. Conclusions

We presented a system for the 3D reconstruction and pose tracking of a near-planar surface in real-time (around 20 frames per second). The system incorporates a simple structured light approach with a pattern of uncoded stripes to acquire 3D surface data, and fits a weighted least-squares plane in order to estimate position and orientation. A possible use was demonstrated in the form of an HCI system where the user controls 3 degrees of freedom of a model aircraft by moving and rotating an open hand in front of the structured light device.

The technique can be applied to various other problems such as the real-time capturing of non-rigid surfaces for deformation analysis or animation purposes, or the data generation and analysis for dynamic scene understanding.

5. Acknowledgments

The author thanks the Council for Scientific and Industrial Research (CSIR) and the Mobile Intelligent Autonomous Systems research group for their support of this work, and Alan Robinson for his useful contributions to the development of the structured light system.

6. References

[1] N.A. Ramey, J.J. Corso, W.W. Lau, D. Burschka and G.D. Hager, "Real-time 3D surface tracking and its applications", Computer Vision and Pattern Recognition Workshop, pp. 34–41, 2004.

[2] S. de Roeck, N. Cornelis and L.J. van Gool, "Augmenting fast stereo with silhouette constraints for dynamic 3D capture", International Conference on Pattern Recognition, pp. 131–134, 2006.

[3] S.Y. Chen, Y.F. Li and J. Zhang, "Vision processing for realtime 3-D data acquisition based on coded structured light", IEEE Transactions on Image Processing, 17(2):167–176, 2008.

[4] S. Zhang and P.S. Huang, "High-resolution, real-time 3-dimensional shape measurement", Optical Engineering, 45(12):123601, 2006.

[5] L. Zhang, B. Curless and S.M. Seitz, "Rapid shape acquisition using color structured light and multi-pass dynamic programming", International Symposium on 3D Data Processing, Visualization and Transmission, pp. 24–36, 2002.

[6] C. Beumier and M. Acheroy, "3D facial surface acquisition by structured light", International Workshop on Synthetic Natural Hybrid Coding and 3D Imaging, pp. 103–106, 1999.

[7] W. Brink, A. Robinson and M. Rodrigues, "Indexing uncoded stripe patterns in structured light systems by maximum spanning trees", British Machine Vision Conference, pp. 575–584, 2008.

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

[9] O. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig?", European Conference on Computer Vision, pp. 563–578, 1992.

[10] Z. Zhang, "A flexible new technique for camera calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330–1334, 2000.

[11] C. Harris and M.J. Stephens, "A combined corner and edge detector", Alvey Vision Conference, pp. 147–152, 1988.

[12] R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision"', 2nd ed., Cambridge University Press, Cambridge, 2003.

[13] R.B. Fisher and D.K. Naidu, "A comparison of algorithms for subpixel peak detection", Advances in Image Processing, Multimedia and Machine Vision, Springer-Verlag, Heidelberg, 1996.