

# IMPLEMENTATION OF A ROBUST CODED STRUCTURED LIGHT TECHNIQUE FOR DYNAMIC 3D MEASUREMENTS

*Jordi Pagès, Joaquim Salvi and Carles Matabosch*

Institute of Informatics and Applications  
University of Girona  
Av. Lluís Santaló s/n, 17071 Girona (Spain)

## ABSTRACT

This paper presents the implementation details of a coded structured light system for rapid shape acquisition of unknown surfaces. Such techniques are based on the projection of patterns onto a measuring surface and grabbing images of every projection with a camera. Analyzing the pattern deformations that appear in the images, 3D information of the surface can be calculated. The implemented technique projects a unique pattern so that it can be used to measure moving surfaces. The structure of the pattern is a grid where the color of the slits are selected using a De Bruijn sequence. Moreover, since both axis of the pattern are coded, the cross points of the grid have two codewords (which permits to reconstruct them very precisely), while pixels belonging to horizontal and vertical slits have also a codeword. Different sets of colors are used for horizontal and vertical slits, so the resulting pattern is invariant to rotation. Therefore, the alignment constraint between camera and projector considered by a lot of authors is not necessary.

## 1. INTRODUCTION

Stereovision is one of the computer vision techniques that can be used to obtain 3D information of the environment. The stereovision principle consists of imaging a certain surface from two or more cameras. Then, solving the correspondence problem between cameras, triangulation can be used to reconstruct the 3D position of the matched pixels. However, the correspondence problem is not always easy to solve. In order to ease the problem structured light techniques appeared. Such techniques are based on replacing one of the cameras by a light source. Then, projecting a set of known patterns onto the measuring scene and grabbing images with the remaining camera(s), the correspondence problem is solved by all those points where the patterns have been projected on.

A structured light system can achieve some important features like: large number of 3D reconstructed points (high

resolution) well distributed over the measuring surface and good accuracy in the 3D measurements. Most part of structured light systems are based on scanning a laser stripe onto the measuring surface and grabbing images for every known angle of the laser source. Such systems obtain very accurate measurements, but the large number of images that must be grabbed does not permit to measure moving surfaces. In order to solve such limitation, some coded structured light systems mainly based on a unique pattern projection can be used [1]. Such systems are coded by means of spatial neighborhoods.

Spatial neighborhood codification consist of identifying a set of points of the pattern with the information contained in a small neighborhood around each one of them. Then, a set of well distributed pixels on the image can be reconstructed with a single projection. The unique drawback of such technique is that, since the codification must be condensed in a unique pattern, the spatial resolution that can be achieved decreases.

This paper presents the implementation of a low-cost coded structured light technique based on spatial neighborhood codification. The resulting system permits to obtain the shape of an unknown surface even if it is moving. Besides, since the proposed pattern has a grid structure, both row and column codification is included into the crossing points. With such redundancy some interesting features are obtained: high accuracy in the reconstruction of the cross points, possibility to reconstruct both vertical and horizontal slits, and error detection and correction when decoding the cross points.

The paper is structured as follows: first, in section 2 the hardware required for the technique is summarized. Secondly, the structure of the pattern is detailed in section 3. The image processing and the decoding stage of the pattern observed by the camera are presented in section 4. Afterwards, in section 5 some examples of surface reconstruction using the implemented technique are shown. The paper ends with conclusions.

*Work funded by Spanish project CICYT TAP99-0443-C05-01*

## 2. IMPLEMENTING A LOW-COST SYSTEM

The implementation of a low-cost structured light system is achieved with a standard LCD projector, a camera, a frame grabber and a PC. The LCD device permits to project bidimensional patterns defined as images.

The devices used for our implementation are: a Mitsubishi XL1 video projector, a Sony 3 CCD camera, a Matrox Meteor-II frame grabber and a standard PC. The resolution of the projected pattern is  $1024 \times 768$  pixels, while the camera images are digitized at  $768 \times 576$  pixels, with a depth of 24 bits per pixel.

## 3. DESIGN OF THE PATTERN

The pattern used for the current implementation was proposed by Salvi et al. [4]. The structure of the projected pattern is defined as a grid composed of vertical and horizontal colored slits of a certain thickness over black background. In order to choose the sequence of colors that is assigned to the horizontal and vertical slits, a De Bruijn sequence is used. De Bruijn sequences have been used by several authors with the aim of defining patterns without periodicity. Concretely, the most exploited patterns coded with De Bruijn sequences consists of parallel slits, i.e. the one proposed by Monks et al. [3].

A De Bruijn sequence of order  $m$  over an alphabet of  $n$  symbols is a circular string of length  $n^m$  that contains each substring of length  $m$  exactly once. Such characteristic is called window property. For example, given the De Bruijn sequence of eq. 1, if every element of the alphabet  $\{0, 1, 2\}$  is mapped to a certain color, a total of  $3^3 = 27$  parallel slits of a pattern can be colored by mapping such sequence. The given sequence has length 29 due to its circular property.

$$220210200120111010002212111222 \quad (1)$$

Since the given De Bruijn sequence has a window property of 3, every three consecutive slits in the pattern will be uniquely identified by the codeword formed by their three colors. Another author who took profit of this coding strategy was Zhang et al. [5] proposing a coded pattern containing 125 vertical slits with a De Bruijn sequence of order 5 and 5 colors.

Other pattern structures coded with De Bruijn sequences have been proposed. For example, Griffin et al. used an array of colored dots such that every window of  $3 \times 3$  dots in the pattern was unique [2]. The dot representation requires to locate the mass centers of all the imaged spots in order to triangulate them. Nevertheless, all those spots that appear partially occluded in the camera image must be discarded since their mass center might not be well segmented.

The work by Salvi et al. [4] proposed to design a grid pattern instead of a single sequence of parallel slits or a dot

array. By selecting the colors of both vertical and horizontal slits with the same De Bruijn sequence, both axis are coded. The set of colors used to encode vertical and horizontal slits is different in order to differentiate both kind of slits. All the pixels belonging to a vertical or horizontal slit have a codeword which indicates the position of the slit in the pattern. All the imaged pixels belonging to the slits can be reconstructed by intersecting the equation of the camera 3D line which contains the image pixel and the equation of the 3D plain corresponding to the slit. Moreover, the cross points of slits in the grid have two codewords. Therefore, cross points can be reconstructed more accurately by intersecting the equations of two 3D lines. Another advantage of this pattern structure is that redundancy in the coding is included since two codewords are defined for every cross point. This fact permits to detect and even correct errors in the decoding stage of the imaged pattern.

The colors used for the horizontal slits are red, green and blue, while yellow, cyan and magenta are used for the vertical slits. The resulting pattern can be observed in figure 1.

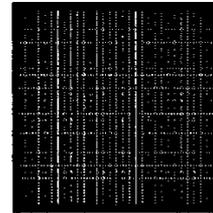


Fig. 1. Grid pattern of  $29 \times 29$  slits using red, green and blue for horizontal slits and yellow, cyan and magenta for vertical slits.

## 4. PATTERN DECODING

Once the pattern is projected onto the measuring surface, an image must be grabbed with the camera. Then, for all those cross points of the grid that can be identified and decoded, the correspondence problem between camera and projector can be solved and, therefore, the corresponding 3D points can be triangulated. In order to segment the grid in the image, a stage of image processing must be fulfilled. Then, the decoding stage must obtain the codewords of every visible cross point of the grid. This stage must be robust to errors since some parts of the projected grid can be occluded from the camera point of view.

The implemented algorithm has been structured in the following steps: first the segmentation of the grid, then, the cross points detection, and finally, the decodification of the detected cross points. Hereafter, all three steps are detailed.

#### 4.1. Segmentation of the grid

For correctly segment the projected grid it is necessary to clearly distinguish the 6 primary colors used. In order to success, a color calibration procedure is made only once, when installing the system in the working area, by projecting the grid onto a color-neutral surface and calibrating the gains for every projected color. The camera iris is also adjusted to perceive basically only the projected grid.

The first step of the segmentation algorithm consists of applying a Sobel operator to the camera image in order to detect the edges of the projected grid. The resulting image is converted to 8-bit greyscale. Afterwards, three Close morphological operators are applied in order to merge the parallel edges that appear using Sobel filter. Then, the obtained image is binarized with a low threshold in order to get thick slits. Finally, a thinning algorithm must be applied until the skeleton of the image is obtained. The sequence of operations can be observed in figure 2.

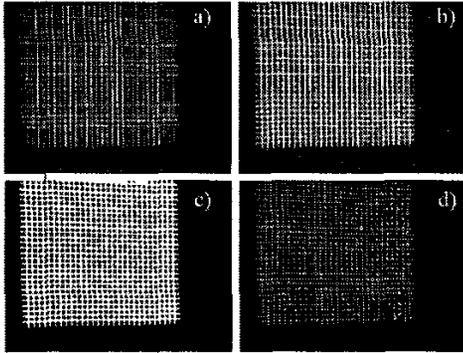


Fig. 2. Grid segmentation. a) original 24-bit image. b) image after edge detection, conversion to 8-bit and 3 Close iterations. c) binarization. d) thinning until skeleton.

#### 4.2. Cross points detection

Once the camera image has been processed in order to enhance and to segment the grid skeleton, the cross points between horizontal and vertical slits can be located. The set of masks shown in figure 3 are convolved with the image containing the skeleton of the grid.

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \dots$$

Fig. 3. Samples of the binary masks used to detect grid cross points.

For every position of the image where the convolution

result is greater than 6, the pixel is considered to be a cross point of the grid. All these masks must be used since the thinning operation to obtain the grid skeleton not always leads to perfect intersections.

An example of the process can be seen in figure 4, where the pattern has been projected onto a horse statue.

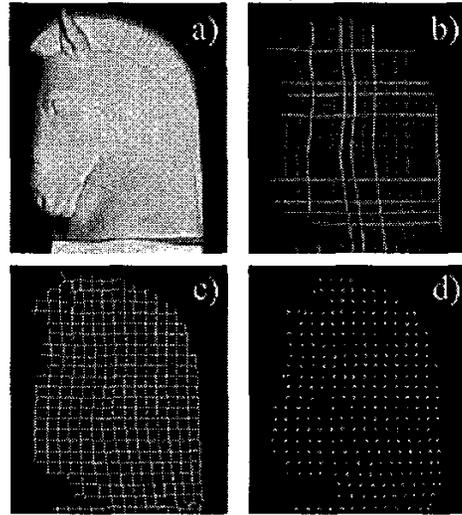


Fig. 4. Cross points detection. a) A horse statue. b) the pattern projected. c) the extracted skeleton of the grid. d) the detected cross points.

#### 4.3. Decoding the detected cross points

The projected grid is colored so that every cross point has two codewords. The first codeword is formed by the colors of the vertical slit containing the cross point and the colors of both adjacent vertical slits. The second is generated in the same way but with the horizontal slits. Both codewords are unique for every cross point (window property), so there is a direct mapping from both codewords to the cross point coordinates in the pattern.

In the previous subsection the process to detect cross points in the camera image has been explained. In order to decode the maximum number of cross points the following algorithm is applied:

- Generation of the cross points adjacency graph
- Obtaining the horizontal and vertical slits colors corresponding to every cross point
- Decoding cross points by using both adjacency graph and colors of slits
- Transfer of codewords to neighbors not decoded
- Correction of inconsistent codewords

To generate the adjacency graph is necessary to start from the coordinates of every cross point and track the edges of the grid skeleton towards four directions in order to find the neighbors. Since the skeleton has thickness of 1 pixel, windows of dimensions  $1 \times 3$  and  $3 \times 1$  are enough for tracking vertical and horizontal edges respectively.

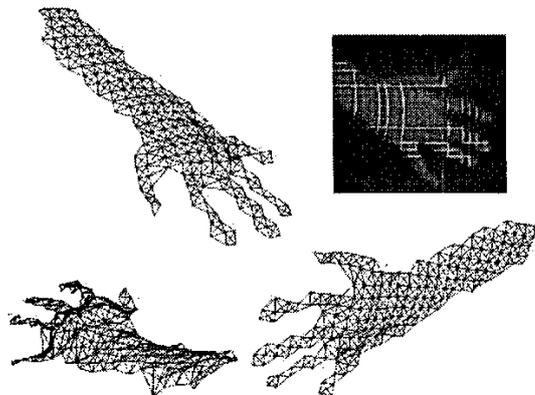
Once the colors of the slits intersecting in every cross point are found, the codewords can be obtained for all those cross points whose four neighbors have been located. Otherwise, the cross points are not decoded.

The following step tries to decode such cross points not decoded in the previous step by choosing the proposed codeword most voted by all the available decoded neighbors. For example, a cross point whose decoded vertical slit has index 12 will propose the codeword 13 to its right neighbor.

The final step of the algorithm consists of comparing the codewords of every cross points with the codewords that its neighbors would propose. Then, if the codewords of a cross point do not coincide with the most voted codewords proposed by its neighbors, such most voted codeword will be accepted as the correct one. In the same way, if a cross point only has a neighbor, the codeword proposed by the neighbor is directly adopted. The last step of the algorithm is repeated until no changes in the codewords occur.

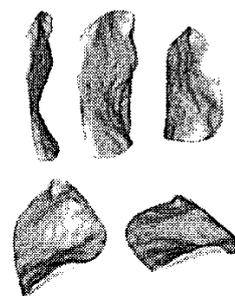
## 5. EXAMPLES OF RECONSTRUCTION

After the decoding stage, all those cross points that have been decoded can be triangulated obtaining a cloud of 3D points which fit on the measuring surface. In figure 5 the reconstruction of a human hand is presented.



**Fig. 5.** Reconstruction of a hand: the hand under the projected pattern and different views of the reconstructed triangle mesh.

The profile of the horse statue from figure 4 can be observed in figure 6.



**Fig. 6.** Reconstructed surface of the horse.

## 6. CONCLUSIONS

The implementation of a coded structured light technique based on the projection of a grid pattern has been detailed. The paper focuses on the image processing and proposes a robust decoding stage.

The pattern structure improves other pattern representations proposed in the bibliography. The different set of colors used for vertical and horizontal slits permit to easily distinguish between them. Since both axis of the pattern are coded, cross points of the grid are redundantly coded, leading to a more accurate triangulation. Redundancy also permits to detect and correct errors in the cross points decodification. The resolution of the technique can be increased by enlarging the number of slits and, therefore, the length of the De Bruijn sequence used to encode the grid. Moreover, since the pixels belonging to the horizontal and vertical slits are also coded with respect to a single axis, they can also be reconstructed producing a denser surface.

## 7. 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*, 31(7):963–982, 1998.
- [2] P.M. Griffin, L.S. Narasimhan, and S.R. Yee. Generation of uniquely encoded light patterns for range data acquisition. *Pattern Recognition*, 25(6):609–616, 1992.
- [3] T. P. Monks, J. N. Carter, and C. H. Shadle. Colour-encoded structured light for digitisation of real-time 3D data. In *International Conference on Image Processing*, pages 327–30, April 1992.
- [4] J. Salvi, J. Batlle, and E. Mouaddib. A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters*, (19):1055–1065, September 1998.
- [5] L. Zhang, B. Curless, and S. M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *Int. Symposium on 3D Data Processing Visualization and Transmission*, Padova, Italy, June 2002.