3D feature tracking using a dynamic structured light system

Conference Paper · June 2005

DOI: 10.1109/CRV.2005.1 · Source: IEEE Xplore

4 authors, including:

Antonio Adan
University of Castilla-La Mancha
129 PUBLICATIONS **1,173** CITATIONS
SEE PROFILE

Andres S. Vazquez
University of Castilla-La Mancha
32 PUBLICATIONS **213** CITATIONS
SEE PROFILE

3D Feature Tracking Using a Dynamic Structured Light System

Antonio Adán, Fernando Molina, Andrés S. Vázquez, Luis Morena
Departamento de Ingeniería Eléctrica, Electrónica y Automática. UCLM
Paseo de la Universidad 4. Ciudad Real. Spain
{Antonio.Adan, Fernando.Molina, Andress.Vazquez}@uclm.es, luismorena@yahoo.es

Abstract

In this paper we present a 3D tracking system that is able to identify several characteristic points on the surface of a moving scene. The system is considered as a preliminary step in the interaction of industrial robots with dynamic scenes. Through a new color structured light technique based on a disordered codeword pattern, 3D coordinates of the object surface are extracted and processed. This method recovers 3D information in such a way that the correspondence problem is easily and robustly solved. After establishing a set of feature points, an inter frame window search algorithm is carried out to solve the tracking problem.

A controlled experimental setup has been built in our lab composed by a 2 DOF mobile platform, a light structured sensor and a manipulator robot. The experimentation has been performed on medium spatial resolution and for soft movement specifications giving promising results.

1. Tracking strategies

Tracking and motion recognition have received an increasing attention in recent years [1, 2]. Indeed, tracking is one of the most important issues in many applications, where the orientation, the position, the motion or the features of an object are required, and a problem of general interest in computer vision. It is also an extensive field which has been classified according to diverse criteria.

From an application point of view, trackers can be broadly categorized into robotics applications, monitoring and augmented reality [3]: more specific applications range from teledetection, security, traffic control, animation, special effects, video compression, medical imaging, agricultural automation.

Regarding its methodology, tracking has been classified in two fields: image-based and model-based tracking [4]. Tracking through image processing is used to detect moving regions, contours, textures, oriented points or optical flows. In [5] three major approaches are considered: feature tracking, where feature points are tracked using a small window, contour tracking, and template tracking, where a small region of the image is tracked using the pixels' intensities. In the model-based methods a model is given to the algorithm before or during the process of segmentation.

Three further categorizations of tracking techniques are of interest when considering whether the solution proposed: a) uses or not 3D data either as a model or as a range image sequence, b) is stable, that is, suitable for applications such as Augmented Reality, c) is real-time. For example, in [6] a stable, real-time solution for tracking rigid objects in 3D using a single camera is proposed that can handle large camera displacements and partial occlusions. However the tracker must start with a small user-supplied set of keyframes and requires a 3D model of the target object or objects. In [7] a technique for human motion capture is described which is marker-free and non-model based; that is, not prior information about the body structure is needed. A kinematic model and its parameters of motion are estimated from a volume sequence reconstructed from multi-view video data by means of a shape-from-silhouette technique. The algorithm is applicable to other moving subjects but limited to those whose structure can be modeled as linked kinematic chain, e.g. animals and mechanical devices.

For tracking objects using exclusively 3D data, variants of the Iterative Closest Point algorithm (ICP) are normally used [8, 9, 10] although other emerging methods include variants of the Hough transform [11]. In [8], Simon et al. describe an approach for full 3D pose estimation of a rigid object at speed up to 10 Hz. A triangular mesh model of the tracked object is generated offline using a conventional range sensor.

Real-time range data of the object was sensed by a high speed range sensor. Pose estimation is obtained by registering the real-time range data to the triangular mesh model and the method does not require explicit feature extraction or specification of correspondence. In 2003 [10], Blais et al. present a recursive optimisation method to produce a high-resolution range image of a free-floating object using sparse range data. A rough, distorted model of the object is created which is recursively optimised using new range information and the ICP algorithm. Real-time tracking of the free-moving object is performed to allow the laser-scan to be automatically centered on the object. In [11] a variation of the General Hough transform is introduced to track objects in a sequence of sparse range images. The method makes use of the coherence across image frames that results from the relationship between known bounds on the object's velocity and the sensor frame rate.

In general, this type of techniques, which only use 3D data, can register moving surfaces because a range image is obtained from a single frame. Although the spatial resolution in moving applications is usually low, it may be sufficient for feature tracking.

The motivation behind our research is addressed to the interaction of industrial manipulators with moving objects and so, in this paper, we are mainly concerned with the problem of 3D tracking using range image sequences. A remarkable difference with respect to previous approaches is that instead of tracking an object we track a set of feature points of a free moving-object. This involves computing a set of 3D coordinates for each frame of the movement. To do that, we have designed a range sensor which is appropriate for shape recovering of dynamic scenes.

This paper is structured as follows. Section 2 briefly describes the structured light system that we have designed for synthesizing moving 3D shapes. Section 3 explains the proposed method of tracking. Section 4 presents a real tracking experimentation and Section 5 suggests directions for future work.

2. A structured light system for dynamic scenes based on disordered codewords

2.1. Introduction

It is well known that structured light techniques are a good alternative to the classical stereovision techniques for surface registration applications in dynamic environments. In this sort of techniques, one camera of the stereo system is substituted by a device that projects light patterns on the scene. Salvi et al. have recently presented a complete overview in [12].

Time-multiplexing strategies, where a set of patterns are successively projected, are used for static scenes. On the contrary, strategies that project a single pattern are suitable for dynamic scenes. The essential difference among dynamic methods consists on how the nodes (or points) of the pattern are encoded and the way of making the camera-projector correspondence. In this section, we briefly present a new strategy for recovering 3D surfaces in moving environment. A more complete report about our system can be found in [13].

It can be said that strategies based on neighborhood are the most popular and effective encoding techniques. In these, a point of the pattern is encoded taking into account the position and the features (symbols) of a set of points that are generically called its 'neighbors'. The encoding methods based on De Bruijin sequences constitute a group of techniques that define patterns well-built in a mathematical sense [14]. Most of them use a one-dimensional sequence of n symbols in which all possible windows of l-dimension ($l<<n$) are unique on it. This idea has been directly applied to encode rows or columns in patterns.

Strategies that identify points/spots instead of axes are also very efficient. In these methods, it is common to design a map of features or symbols in the two-dimensional space with 4/8 connectivity and where each two-dimensional window appears once. If all possible windows are included in the pattern, its structure is called *perfect map* or *perfect submap* otherwise. These techniques also differ in the way the encoding is inserted in the pattern. In this field, works by Griffin et al. [15], Davies et al. [16] and Morano et al. [17] stand out.

2.2. Disordered perfect submap

Our strategy may be included in the *perfect submaps* category. The pattern designed consists of a *perfect submap M* with 6-connectivity topology where a window w is defined as a set of seven symbols that verify the followings properties:

1. w appears once in the pattern.
2. w can contain repeated symbols.
3. The order in w is irrelevant.

It can be said that our pattern has been inspired on strategies based on two dimensional array patterns but with a different topology. A complete report about our system can be found in [13].

In order to design a map of features (symbols) on M, we have used seven colors (red, green, blue, magenta, cyan, yellow and white) associating one color to each point of M. From now on, we will denote C the color that has been associated to the point c.

Consequently a codeword corresponding to point c consists on the pair $[C, V(C)]$ where the topology and properties for points remain for colors. The color has been inserted in M by applying an iterative algorithm that starts with a random assignment and that converges when no repeated codes are found. That is how we have generated such a pattern establishing a codeword as a set (in the mathematical sense) of elements. Therefore, for us, [*b*, *bymymg*] put in the Figure 1 is equivalent to [*b*, *gbyymm*].

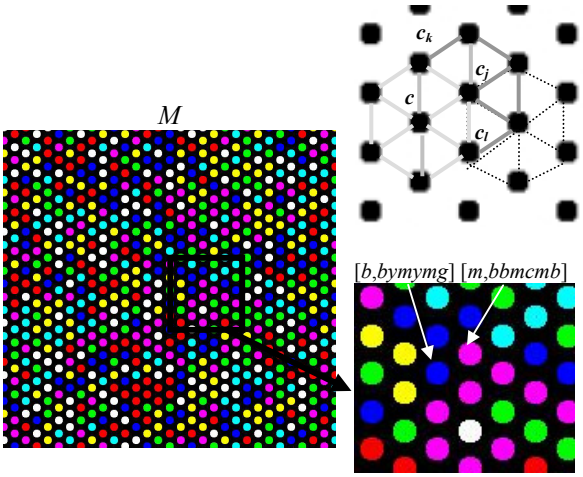


Figure 1. Color disordered pattern

2.3. System description

A brief description of our structured light system is as follows. A spot C corresponding to a point c of M, with pattern coordinates (x_c, y_c), is projected in the 3D space on a point w, with world coordinates (x_w, y_w, z_w). The scene is captured by the camera and the point c is registered at image coordinates (x_i, y_i). In order to recover (x_w, y_w, z_w) through (x_c, y_c) and (x_i, y_i) two types of calibrations have been carried out: camera calibration and light projector calibration (Fig. 2).

Firstly, a usual camera calibration produces a relationship between world and computer (pixel) coordinates through a camera calibration matrix C_i.

$$\begin{bmatrix} \alpha x_i \\ \alpha yi \\ \alpha \end{bmatrix} = C_i \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad C_i = \begin{bmatrix} A_{111} & A_{112} & A_{113} & A_{114} \\ A_{121} & A_{122} & A_{123} & A_{124} \\ A_{131} & A_{132} & A_{133} & A_{134} \end{bmatrix} \qquad (1)$$

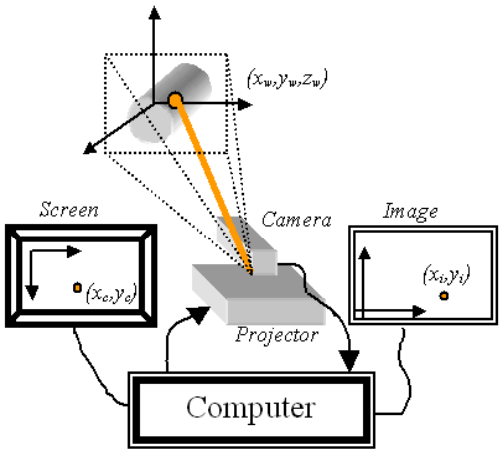


Figure 2. Reference systems

Secondly, the calibration for the projection device is accomplished. Next equation (2) is solved and a second calibration matrix C_c is obtained.

$$\begin{bmatrix} \beta x_c \\ \beta y_c \\ \beta \end{bmatrix} = C_c \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \quad C_c = \begin{bmatrix} A_{211} & A_{212} & A_{213} & A_{214} \\ A_{221} & A_{222} & A_{223} & A_{224} \\ A_{2131} & A_{232} & A_{2133} & A_{234} \end{bmatrix} \qquad (2)$$

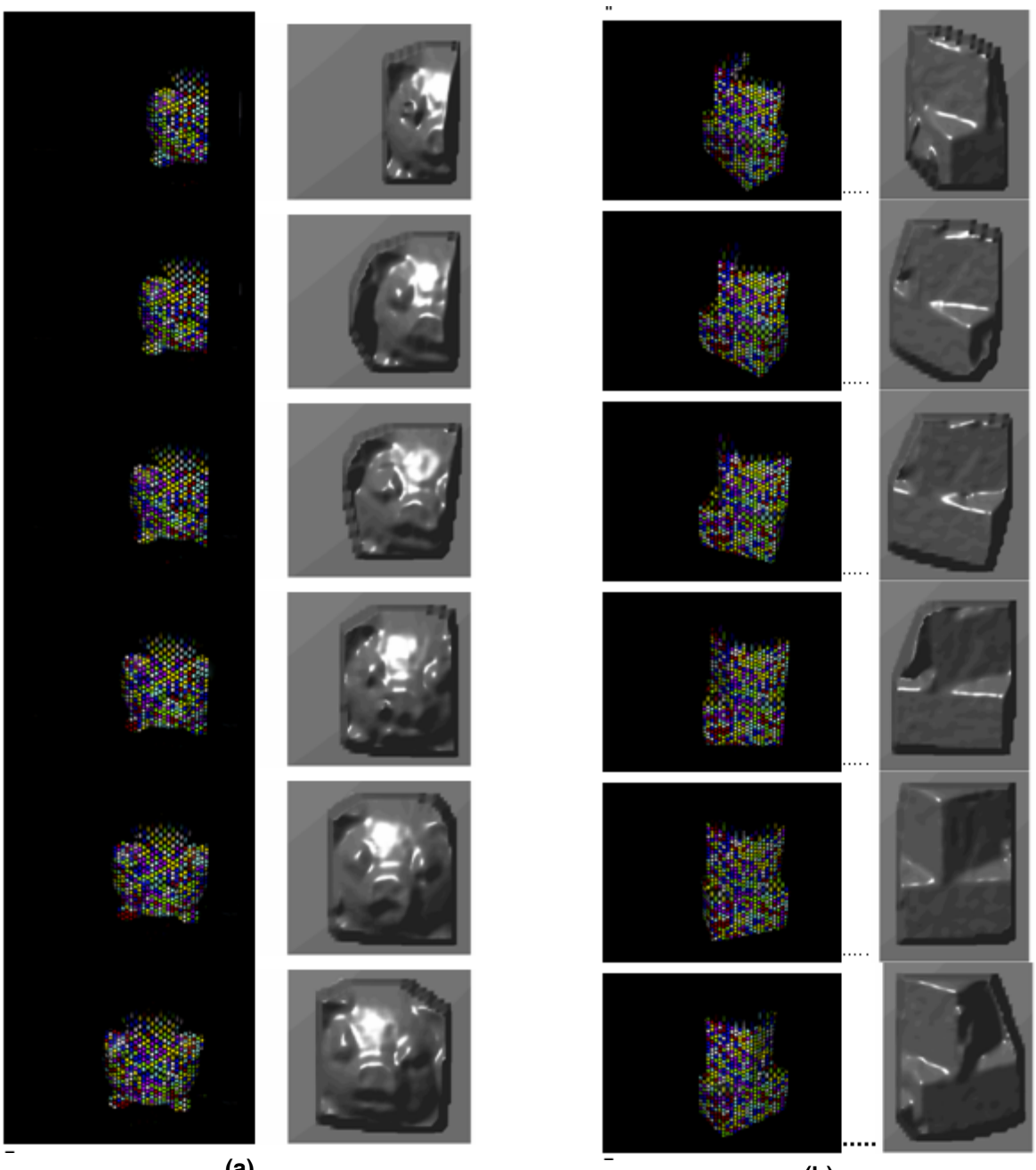
If C_i and C_c are known and their corresponding coordinates (x_i, y_i) and (x_c, y_c) are previously put in correspondence, (x_w, y_w, z_w) can be obtained for every point of the scene using equations (1) and (2). After several manipulations of them, we obtain the following expression:

$$\begin{bmatrix} x_w & y_w & z_w \end{bmatrix}^T = (A^T A)^{-1} A^T B \qquad (3)$$

where:

$$A = \begin{bmatrix} A_{111} - A_{131}x_i & A_{112} - A_{132}x_i & A_{113} - A_{133}x_i \\ A_{121} - A_{131}y_i & A_{122} - A_{132}y_i & A_{123} - A_{133}y_i \\ A_{211} - A_{231}x_c & A_{212} - A_{232}x_c & A_{213} - A_{233}x_c \\ A_{221} - A_{231}y_c & A_{222} - A_{232}y_c & A_{223} - A_{233}y_c \end{bmatrix} \quad B = \begin{bmatrix} A_{134}x_i - A_{114} \\ A_{134}y_i - A_{124} \\ A_{234}x_c - A_{214} \\ A_{234}y_c - A_{224} \end{bmatrix}$$

Concerning the pattern, we have designed a pattern consisting of 986 color spots disposed as a hexagonal array. Figure 3 shows details of 3D reconstructions for two moving objects. In case a) the scene is translated whereas in b) is rotated in front of the camera. Rendering representations and images with the projected pattern are displayed for several frames of a sequence. As it can be seen, an acceptable 3D shape recovering is achieved.



(a) (b)

Figure 3. Image of the scene and 3D reconstruction for two moving scenes

2.4. Pattern-image correspondence

The main property of our system is that the correspondence between image spots and pattern spots coordinates is easily solved. Since the codeword is defined allowing repetition and disorder, several important advantages can be pointed out. First, the average Hamming distance computed in the pattern meaningfully increases being $\overline{H} > 3$. So this encoding is more robust to noise, shades and occlusions conditions. It has been computed, for several implemented patterns, as follows:

$$\overline{H} = \frac{1}{6m} \sum_{\forall C \in M} \left[\sum_{\forall C_j \in V(C)} \left[H(C, C_j) + H(V(C), V(C_j)) \right] \right] \quad (4)$$

where we consider that each codeword has got six neighbor codewords and six distances can be computed. Since 95% of the codes have a Hamming distance greater than one, most of misinterpreted and lost codewords can be corrected and the number of unlabeled spots decreases. Second, the code recovering and the correspondence process are simplified because of the codeword disorder property. For example, tasks such as computing the angle of $V(C)$ with respect to C do not need to be carried out thus reducing the computational cost.

When the color pattern M is projected on the scene, a CCD camera captures a frame sequence. Then, for each frame, the spots are segmented, their coordinates obtained and their color read. Finally, the set B of codewords of the form $[C, V(C)]$ is obtained for each frame.

The specific correspondence problem is solved by finding the common codewords in M and B. Next a checking phase is performed to correct and recover misinterpreted and lost codes, which can be mainly generated by two reasons. Firstly, a spot may be absent in the image due to depth discontinuities or it may be a misinterpreted color due to shaded conditions. Secondly, spots belonging to the edges of the object image are obviously lost codes because they are incomplete (Fig. 4).

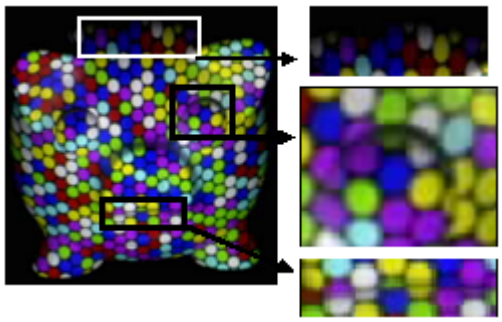


Figure 4. Example of misinterpreted color (up) and lost spots (down)

Fortunately, since we have designed a perfect submap where the Hamming distance is frequently greater than one, most of them can be recovered and matched. A more detailed description of the correspondence process can be found in [13].

3. Tracking of feature points

The proposed method estimates time sequences of 3D points for tracking features of an object as follows. Once the color pattern is projected on the scene, a CCD camera captures a frame sequence of a moving, isolated object and the color spots are read for each frame. The final output is a structure of 3D point clouds belonging to the object surface.

Now we present how the selection of feature points and the analysis of their type are carried out. Firstly n points of the first range image (usually one up to three for the objects tested) are determined automatically, by finding local extrema, or manually. This second option has been used in the experiments below: by positioning the cursor using a mouse, each point results from the intersection of the surface fitted to the point cloud and the selection ray. Given the coordinates of each point we build a small window and interpolate a square grid for the 3D points of the cloud that fall within the window (Fig. 5).

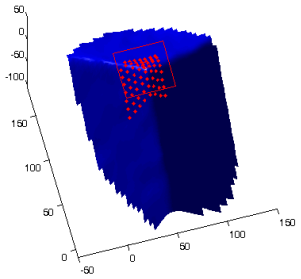


Figure 5. Interpolated grid

In order to know the surface type where the feature points are we compute their Gaussian (K) and mean (H) curvatures and search for extreme values. Note that this selection can also be made automatically by computing directly for the 3D point cloud the local extreme curvature. These curvatures at a vertex point of the interpolated grid are defined from its adjacent triangles according to the respective formulas [18, 19, 20]:

$$K = \frac{\rho \Delta\theta}{A}, \quad \Delta\theta = 2\pi - \sum_i \theta_i, \quad \text{and} \quad A = \sum_i A_i \quad (5)$$

where A is the total area of the adjacent triangles T_i, for $i = 1,..,4$ and ρ is a constant, and

$$-H\vec{n} = \frac{1}{4A} \sum_{j \in N(i)} \left(\cot\alpha_j + \cot\beta_j\right)\left(P_j - P_i\right) \quad (6)$$

where $N(i)$ is the vertex P_i's adjacent polygon set, $(P_j - P_i)$ is the edge e_{ij}, α_j and β_j are two angles in $(j + 1)^{th}$ and $(j - 1)^{th}$ element in $N(i)$ opposite to the edge e_{ij}, respectively, and A is the sum of the areas of triangles in $N(i)$.

Next their surface types can be analyzed. The surface type of a vertex is normally classified using Besl and Jain in [21],

$$T = 1 + 3(1 + sgn(H, \varepsilon)) + (1 - sgn(K, \varepsilon)) \quad (7)$$

where *sgn* is a tolerance signum function. However, as we intend to discriminate only among peak, pit or saddle, we use the simpler formula

$$T = 2*(2+sgn(K))+sgn(H) \qquad (8)$$

where the function *sgn* returns 1 if the curvature is positive and –1 if the curvature is negative. This formula assigns 5, 7 and 1 (or 3) to the surface points peak, pit or saddle respectively. The output of this process will be a selection of feature points for the first frame together with their surface type.

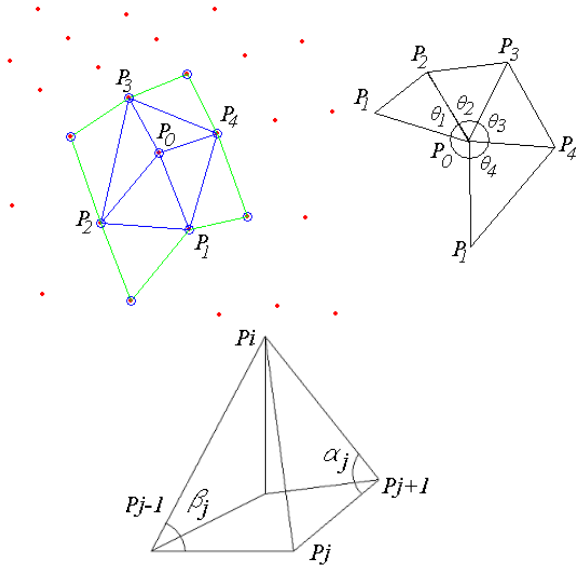


Figure 6. Approximation of Gaussian curvature and mean curvature

Finally feature point correspondences between consecutives frames are found. The coordinates of each feature point allow us to build a small window in the next frame where the correspondent point is expected to fall into, as frame-to-frame disparities are small (Fig. 7). This is possible by acquiring a sequence at a high enough frequency.

For the 3D points that fall within the window we interpolate a grid. And finally for each vertex of this grid we compute its Gaussian and mean curvatures and search for extrema. Notice that this process could have been done directly over the subset of the 3D point cloud within the window but experimental results yield a better approximation when a grid is interpolated.

For example, a point *p* of surface type peak in frame *i*, which produces the window *w* in the consecutive frame *i*+1, is matched to the point *q* of *w* which has maximum Gaussian curvature and negative mean curvature. Similarly, a point of surface type pit in a window *w* is one of maximum Gaussian curvature and positive mean curvature, while a point of surface type saddle is one of minimum Gaussian curvature. This process is illustrated in Figure 8 where two clearly distinctive points of a male head manikin are tracked between consecutive frames.

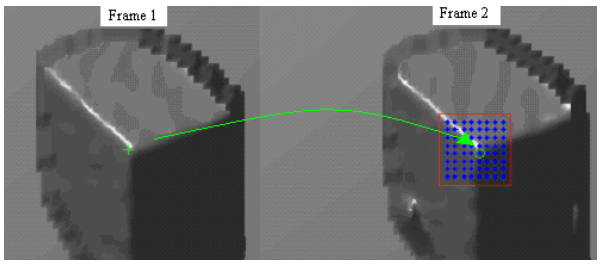


Figure 7. Correspondence process

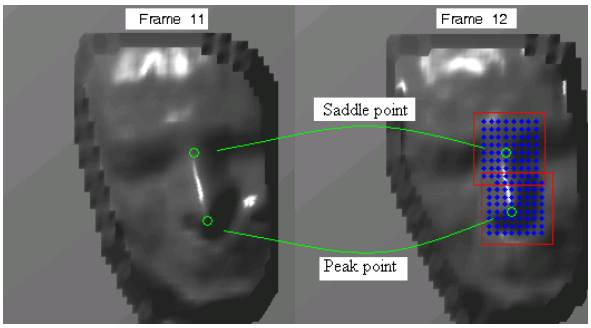


Figure 8. Correspondence process

In this way we build an initial time sequence of 3D points by aggregating all feature points between consecutive frames to the initial selection of frame 1.

4. Experimental test

We have tested our method with real data collected from the range sensor presented in section 2. The experimental setup consists on a XC-003P CCD Sony camera, a MP8755 3M projector, a standard PC computer with a MATROX CORONA frame grabber and an external analog color monitor connected to the camera (Fig. 9). The camera stands 25 cm vertically above the projector. Furthermore a robot Staubli RX90B CS8 with 6 DOF is included in the setup for interaction purposes.

To have a controlled light environment the whole set-up is covered by a black blind. The scenes used in the experiment are given a mat finish with white paint to obtain suitable reflectance conditions. In this environment, we have tested the method on a wide kind of scenes, usually isolated objects, of around 20 cm dimension, that are moved by means of a turntable

and a conveyor belt. This allows us to have 2 DOF and combine rotation (0-360º) and translation movements (0-50cm). At this first stage of our technique, soft movements of 40º/s and 5 cm/s have been registered. Note however that the speed of objects is not an issue since frames could be retrieved from a video sequence as long as there is no blur frames.

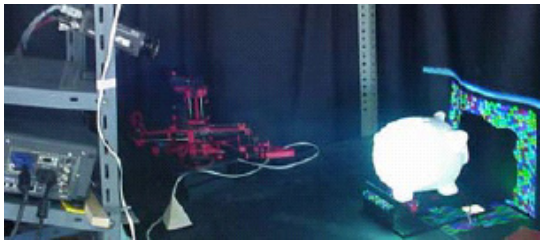


Figure 9. Experimental setup

The pattern size has been set taking into account the size of the scene, the conditions of the experimental set-up and our experimentation requests. With this configuration we extract around 2 points per square centimeter on surfaces 100 cm far away.

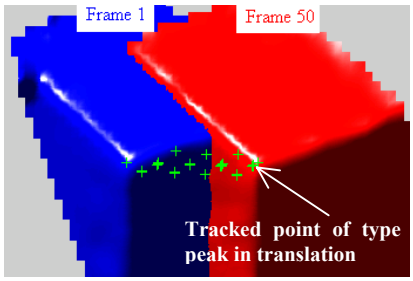


Figure 10. Tracking of a prism's vertex

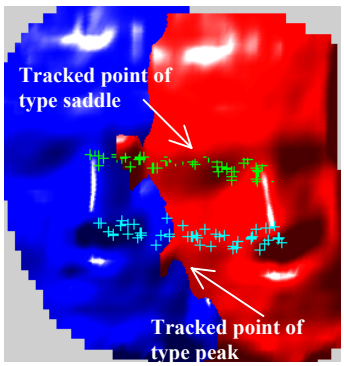


Figure 11. Tracking of a human nose

Above we show details of feature tracking for two sequences taken of two objects: a polyhedral object and a male head manikin. We took 50 and 60 frames respectively in the absence of occlusions. The objects moved at about 5 cm/s which meant a displacement between frames lower than the resolution of the projected pattern. Observe this fact illustrated below where crosses representing the tracked feature tend to accumulate along consecutive points of the hexagonal projected pattern.

In order to validate our tracking method, a robot is inserted in the experimentation setup. Firstly the vision system and the robot coordinate systems are put into correspondence. After finishing the movement synthesis for a test of k frames, registering the frames times and computing the feature points positions, we obtain a sequence of four component vectors $(x,y,z,t)_i$, i=1,..k, for each feature point. Next identical movement of the scene is repeated putting the end-effector of the robot according to each computed location/time of the sequence. Note that, in this case, lightning control is not required. In this experience, errors between real and computed locations can be measured using the coordinates given by the robot. After carrying out a wide experimentation with several objects and movements the results yield the mean errors Δx=3.40 mm, Δy=5.47 mm, Δz=0.65 mm corresponding to errors in the robot coordinate system. Figure 12 shows, for a couple of examples, how the end-effector tracks the trajectory of several feature points: the nose point of a head manikin (a) and a vertex of a polyhedral shape (b). Observe here how the robot works in natural lightning conditions. Two frames of this sequence are also shown in more detail to point out both the feature point and the end-effector positions.

5. Conclusions

In this paper we present a 3D tracking system for a set of feature points belonging to a free moving-object. The motivation of this work is addressed to the interaction of industrial manipulators in moving scenes. The points are usually peak, pit or saddle points and are prefixed in the first frame. The tracking problem consists on identifying them and computing their respective 3D coordinates throughout the moving sequence.

To do that, we have designed a range sensor which is appropriate for shape recovering of dynamic scenes. This is a new coded structured light sensor based on a disordered codeword strategy which solves easily and robustly the correspondence problem. This component yields medium spatial resolution of around 5 mm on surfaces 100 cm far away.

The system has been tested in a real environment for 2 DOF soft movements. Tracking validation has

been carried out introducing a 6 DOF manipulator robot inside the experimental setup which is able to measure the feature point location error.

Current works are addressed to improve several aspects of the system: to improve the sensor resolution by changing the projection pattern, to increase the dimension and speed of the movement and to reduce the tracking errors by refining all calibrations processes included in the system.

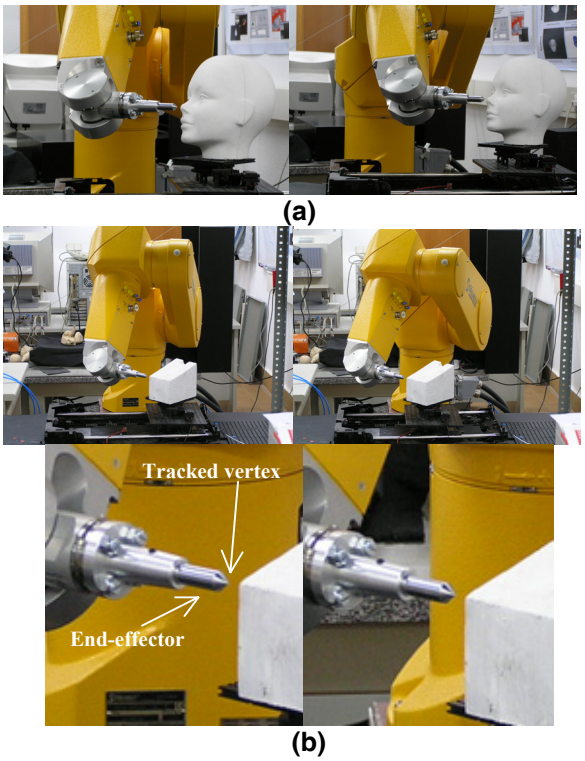


Figure 12. Tracking verification examples

Acknowledgments

This research has been supported by the CICYT Spanish project DPI2001-0986.

References

[1] Shah, J., *Motion-based Recognition*, Kluwer Academic, 1997.

[2] M. Vincze, G. Hager, "Robust Vision for Vision-Based Control of Motion", *Spie*, 2000.

[3] M. Ribo, "State of the Art Report on Optical Tracking", *Technical Report*, VRVIS, 2001.

[4] A. Gagalowicz, P. Gerard, "3D Object Tracking Using Analysis/Synthesis Techniques", *Confluence of Computer Vision and Computer Graphics*, Nato Science Series, pp. 307-329 2000.

[5] E. Bayro, J. Ortegon, "Template Tracking with Lie Algebras", *ICRA*, vol. 5, pp. 5183-5188, 2004.

[6] L. Vacchetti, V. Lepetit and P. Fua. "Stable Real-time Tracking using Online ad Offline Information", *PAMI*, pp. 1385-1391, 2004.

[7] E. Aguiar, C. Theobalt, M. Magnor, H. Theisel and H. Seidel, "M3: Marker-free Model Reconstruction and Motion Tracking from 3D Voxel Data", *12th Pacific Conference on Computer Graphics and Applications*, (PG'04), 2004.

[8] D. Simon, M. Hebert, and T. Kanade, "Real-time 3D Pose Estimation using a High-speed Range Sensor", *ICRA*, pp. 2235-2241, 2004.

[9] P. Jasiobedzki, J. Talbot, and M. Abraham, "Fast 3D Pose Estimation for On-orbit Robotics", *ISR 2000* Montreal, Canada, 2000.

[10] F. Blais, M. Picard and G. Godin, "Recursive Model Optimisation using ICP and Free Moving 3D Data Acquisition", *3DIM*, pp. 251-258, 2003.

[11] M. Greenspan, L. Shang, and P. Jasiobedzki, "Efficient Tracking with the Bounded Hough Transform", *CVPR*, vol. 1, pp. 520-527, 2004.

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

[13] A. Adan, F. Molina, and L. Morena, "Disordered Patterns Projections for 3D Motion Recovering", *3DPVT* 2004, Thessaloniki, Greece 2004.

[14] F.J. MacWillians, and N.J.A. Sloane, "Pseudorandom sequences and arrays", *Proc. of IEEE* 64 (12) 1725-1729, 1976.

[15] P.M. Griffin, L.S. Narasimhan, and S.R. Yee, "Generation of uniquely encoded light patterns for range data acquisition," *International Journal on Pattern Recognition*, 25 (6), pp. 609-616, 1992.

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

[17] R.A. Morano, C. Ozturk, R. Conn, S. Dubin, S.Zietz, and J.Nissanov, "Structured light using pseudorandom codes", *Transactions on Pattern Analysis and Machine Intelligence*, IEEE 20 (3) pp 322-327, 1998.

[18] B. Falcidieno and M. Spagnuolo, "Polyhedral Surface Decomposition Based on Curvature Analysis". *Proc. of the Int. Workshop on Modern Geometric Computing for Visualization*, pp. 263-269. Springer-Verlag, 1992.

[19] L. Zhou and A. Pang, "Metrics and Visualization Tools for Surface Mesh Comparison", *SPIE Proceedings on Visual Data Exploration and Analysis*, Volume number 4302, pp. 99-110, 2001.

[20] M. Desburn, M. Meyer, P. Schröder, and A. Barr, "Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow", *SIGGRAPH99*, pp. 317-324. Addison Wesley, 1999.

[21] P. J. Besl and R. C. Jain, "Segmentation Through Variable Order Surface Fitting", *PAMI*, 10 (2), 1988.