ACTIVE ACQUISITION OF 3D SHAPE FOR MOVING OBJECTS

Marc Proesmans, Luc J. Van Gool and André J. Oosterlinck

ESAT-MI2,
Katholieke Universiteit of Leuven,
Kardinaal Mercierlaan 94,
3001 Leuven, BELGIUM,
E-mail: Marc.Proesmans@esat.kuleuven.ac.be

ABSTRACT

An active 3D acquisition system is presented that projects a simple pattern of squares on a scene and views it from a different angle. This paper describes how the observed pattern can be extracted from the image data. The underlying algorithm automatically detects the lines and crossings of the projected pattern in the image. Experiments show that the algorithm is robust and provides accurate three-dimensional (3D) reconstructions. Its one-shot operation principle enables the system to retrieve the shape of moving objects.

1. INTRODUCTION

Extracting 3D shapes is of increasing interest because of new markets emerging, e.g. virtual and augmented reality. Nevertheless, the acquisition of 3D data is still hampered by the need for expensive equipment and/or the difficulties in its use. ACTS project VANGUARD is all about easing 3D acquisition. The one-shot system presented here lifts several of the current restrictions. It projects a simple pattern of squares onto the scene and extracts the 3D shape data from a single camera image. An example scene is given in fig. 1. Setting up the system is very simple. Of course, detecting the line pattern with sufficient precision is of vital importance and this is the topic of the paper. A description of how the lines yield 3D shape, is given elsewhere [3]. The major advantage of one-shot systems [1, 2, 5, 6] is that they can handle dynamic scenes.

2. DETECTING THE PATTERN

Although the pattern *per se* is of high contrast, its detection should withstand interference from other light sources, surface texture and smearing out by oblique incidence on the surface, and it must also be detected with sufficient precision to obtain good shape reconstructions.

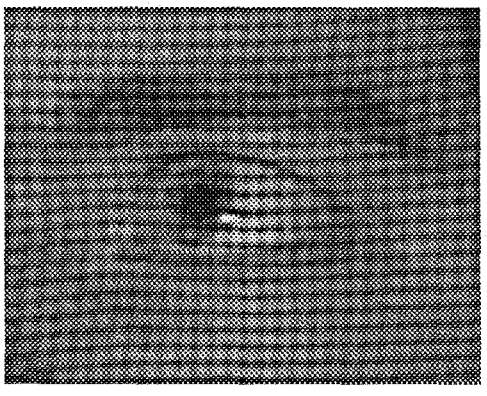

Figure 1: *Projection of a regular square pattern on the scene as observed with a single camera view.*

The assumption made in this work is that both the camera and the projector are put in more or less horizontal positions so that the pattern lines fall into two classes of dominantly horizontal and vertical lines, but note that these restrictions are only loosely imposed. Detecting the lines then proceeds along the following steps:

2.1. Line detector

In a first step, a line detector is used that operates in horizontal and vertical directions separately and localizes minima in intensity. It returns a response that is related to the contrast of the line elements observed in the image. Figures 2b and c show the response of the line detector for the image of fig. 2a.

2.2. Linking line elements

The line detector does not provide connectivity information. Therefore, the individual line pixels are concatenated to form small stripes. Most of the time, gaps may still exist, due to noise or in regions were

0-7803-3258-X/96/$5.00

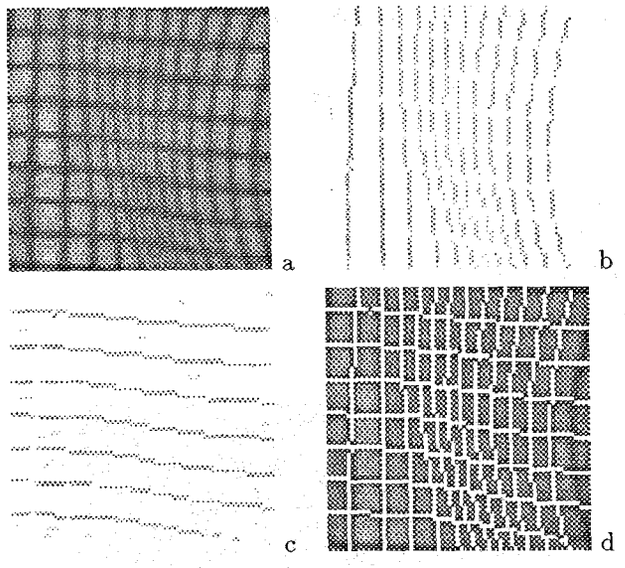

Figure 2: *a: original grey value image of projected pattern. b and c: response of the line detector in vertical and horizontal direction. d: the resulting lines after linking*

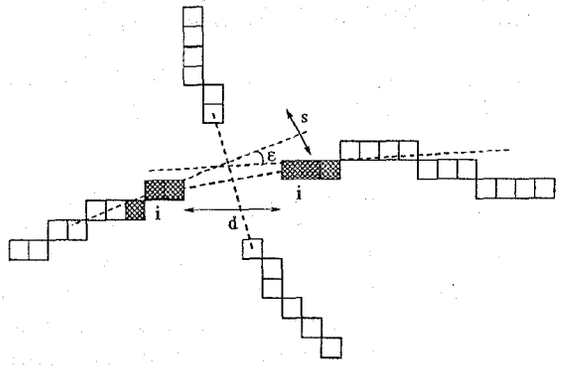

Figure 3: *Linking stripes.*

the line detector's response is poor (e.g. at line crossings). These gaps are bridged by linking the individual stripes. Starting from both ends of each stripe the immediate neighbourhood is scanned over a width s (fig. 3). If points of other stripes are encountered, they are marked as linking candidates. Each candidate is assigned a score based on features such as the difference in angle ϵ between two stripes, the distance between the stripes, the difference in intensity i between end points of the stripe and its candidates, and the mean intensity along the gap, if stripes would be linked. Two stripes will be linked if the sum of their scores maximizes. This results in two sets of lines, one mainly horizontally oriented, the other mainly vertically. Fig. 2d shows the resulting lines after linking.

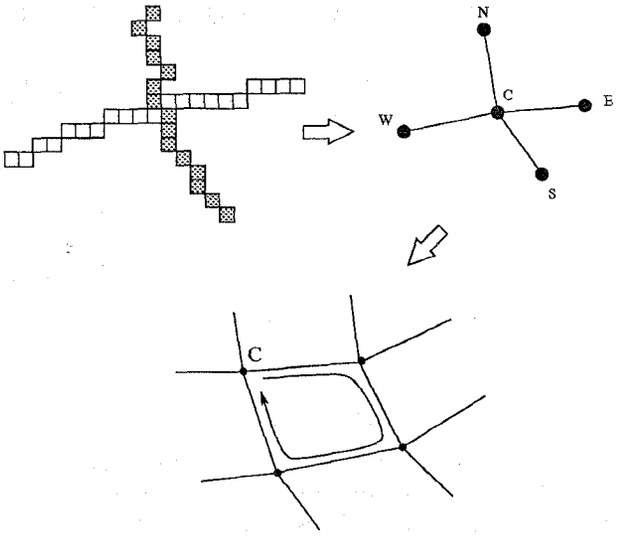

Figure 4: *Constructing the grid.*

2.3. Constructing the grid

The crossings are the locations where both sets of lines meet. Special precautions have been taken since discretized lines do not necessarily contain a common point, or may even have more than one common point at a single crossing. Having determined these, the data is stored in the form of a connected grid. Each grid point has at most 4 neighbours which are labeled N, E, S and W. Grid points that are vertices of complete pattern quadrangles are of special interest, since they indicate how the square pattern is deformed under projection. They can be detected as points C for which the sequence $C \rightarrow E \rightarrow S \rightarrow W \rightarrow N$ brings one back at C.

2.4. Improving accuracy

Pixel accuracy is not enough for good 3D reconstruction. Therefore, the crossings are determined with subpixel precision. A snake-like process changes the grid to let it more closely follow the pattern lines, minimizing

$$\begin{aligned} E &= \sum_{p=1}^{N_E} I(C_p, E_p) + \sum_{p=1}^{N_S} I(C_p, S_p) \\ &+ \lambda\Big(\sum_{p=1}^{N_E}(y_{C_p} - y_{E_p})^2\Big) + \kappa\Big(\sum_{p=1}^{N_E}(x_{C_p} - x_{E_p})^2\Big) \\ &+ \lambda\Big(\sum_{p=1}^{N_S}(x_{C_p} - x_{S_p})^2\Big) + \kappa\Big(\sum_{p=1}^{N_S}(y_{C_p} - y_{S_p})^2\Big) \end{aligned} \quad (1)$$

with N_E and N_S the number of pixels with E and S neighbours, resp., and where $I()$ denotes the average intensity along the line segment between two neighbour-

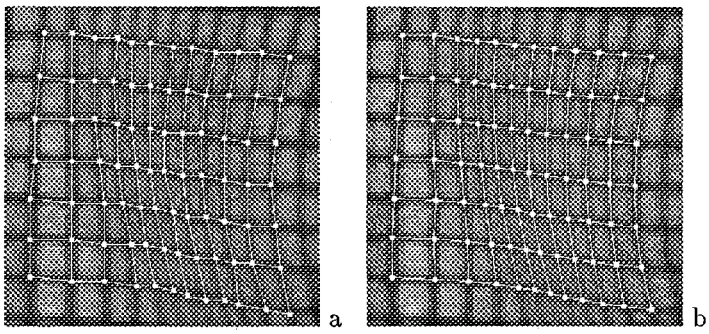

Figure 5: *a: Grid extracted on pixel resolution. b: grid after snake process.*

ing grid points. Minimization moves the grid towards the darker positions in the image, i.e. the line centers. The other terms render the grid smoother. The parameters λ and κ control the smoothness of the lines, where initially $\lambda > \kappa$ (or even $\kappa = 0$), to mainly penalize noise in the portions along the direction normal to the line orientation. Gradient descent is used to minimize E:

$$\forall\, C = (x_p, y_p) \;:\; \frac{dx_p}{dt} = -\nu \frac{dE}{dx_p} \;,\; \frac{dy_p}{dt} = -\nu \frac{dE}{dy_p}$$

Typically, the new crossing positions are much more precise. The effect is shown in figure 5.

2.5. Labeling and detection of inconsistencies and discontinuities

The grid points are labeled according to the line crossing they correspond to. A reference point o is chosen somewhere in the middle of the grid, which will be labeled $(0, 0)$. Starting from this point, the other points are labeled by recursively propagating the labels along the line segments. Depending on the direction of propagation (N, E, S, or W), the vertical and horizontal line labels i and j are incremented or decremented. The labeling information also carries a measure of the reliability of the path of propagation. Due to false connections grid points may have wrong labels temporarily. The reliability of a path depends on the reliability of the points along it, which is a function of the points' connectivity to neighbouring regions. False labels will be overruled by more reliable labels as the process proceeds.

Fig. 6 shows an example where different labels result depending on the chosen path. In this example the paths going upward clearly have to go through points with low connectivity. The large, circumscribing path on the other hand is much more reliable. Its information will overwrite the labels from the upgoing paths.

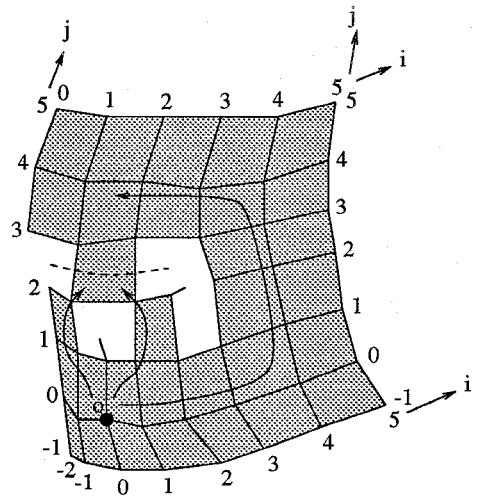

Figure 6: *Labeling of the grid points.*

Line detection will unavoidably introduce false links. For instance, stripes might be linked accross discontinuities, resulting in spurious quadrangles. Noise and tiny details can also introduce errors. The affected quadrangles typically do not satisfy the necessary constraints. The easiest way to verify their validity is to look at the pattern from the projector's view point based on the reconstruction and the extracted relative orientation of the projector [3] (and *not* by taking another image!): quadrangles should look square. The test is based on quadrangle area A over perimeter L squared:

$$\rho = \frac{16A}{L^2} \;,\quad 0 \le \rho \le 1$$

The ratio ρ is optimal ($\rho = 1$) for perfect squares. The more a quadrangle deviates from the square shape, the more ρ will tend to zero. This ratio is introduced to steer the labeling process. The reliability of the path will therefore not only depend upon the connectivity of the grid but also the squareness of the quadrangles as "seen" from the projector. As a result, inconsistencies in labeling can be found nearby gaps or discontinuities in the grid. Gaps will automatically be filled, discontinuities show up as holes in the reconstruction.

Once grid consistency has been restored throughout, the snake process (1) is rerun with $\lambda = \kappa$. This step further reduces the disturbing influence of surface texture.

3. EXPERIMENTS

For the experiments we used a slide projector and a camera with zoom lens. The pattern has been generated using a lithographic pattern generator: $10\mu m$

thick lines at a distance of $50\mu m$. The lens of the camera is set to maximal zoom (focal length about $75mm$). The distance between the projector/camera and the objects was between 3 and $4m$. Fig. 7 shows the reconstruction of a talking face. A discontinuity has been detected at the left side of the nose, which is occluded to the camera. Also note that the results are shown with texture mapping: not only 3D shape but also the intensity information can be obtained as if the lines were "removed" from the original image [4]. We believe this kind of dynamic scene to be of particular relevance because of its usefulness for videotelephony and –conferencing research.

4. CONCLUSIONS

This paper discussed the use of a one-shot system where a particularly simple type of pattern, i.e. a square grid is projected onto the scene. We showed that it is possible to extract accurate grid information and achieve high quality 3D shape information. As was also illustrated, the system is able to retrieve the shape of dynamic scene content.

`http://www.esat.kuleuven.ac.be/~konijn/vanguard.html`

Acknowledgements: Support by ACTS project AC074 "VANGUARD" and a postdoctoral grant for M.P. by the Flemish Inst. for Scient. Techn. Res. in Industry (IWT) are gratefully acknowledged.

5. REFERENCES

[1] A. Blake, D. McCowen, H. R. Lo, and P. J. Lindsey, Trinocular Active Range-Sensing, IEEE PAMI 15(5), pp. 477-483, 1993.

[2] M. Maruyama, and S. Abe, Range Sensing by Projecting Multiple Slits with Random Cuts, IEEE PAMI 15(6), pp. 647-650, 1993.

[3] M. Proesmans, L. Van Gool and A. Oosterlinck, One-Shot Active 3D Shape Reconstruction, accepted for Proc. ICPR, 1996.

[4] M. Proesmans and L. Van Gool, OSIRIS – A system that extracts both range and intensity, Technical Report, ESAT-MI2, Katholieke Universiteit Leuven.

[5] P. Vuylsteke and A. Oosterlinck, Range Image Acquisition with a Single Binary-Encoded Light Pattern, IEEE PAMI 12(2), pp. 148-164, 1990.

[6] M. Watanabe and S.K. Nayar, Telecentric optics for computational vision, 4th ECCV, pp. 439-451, 1996.

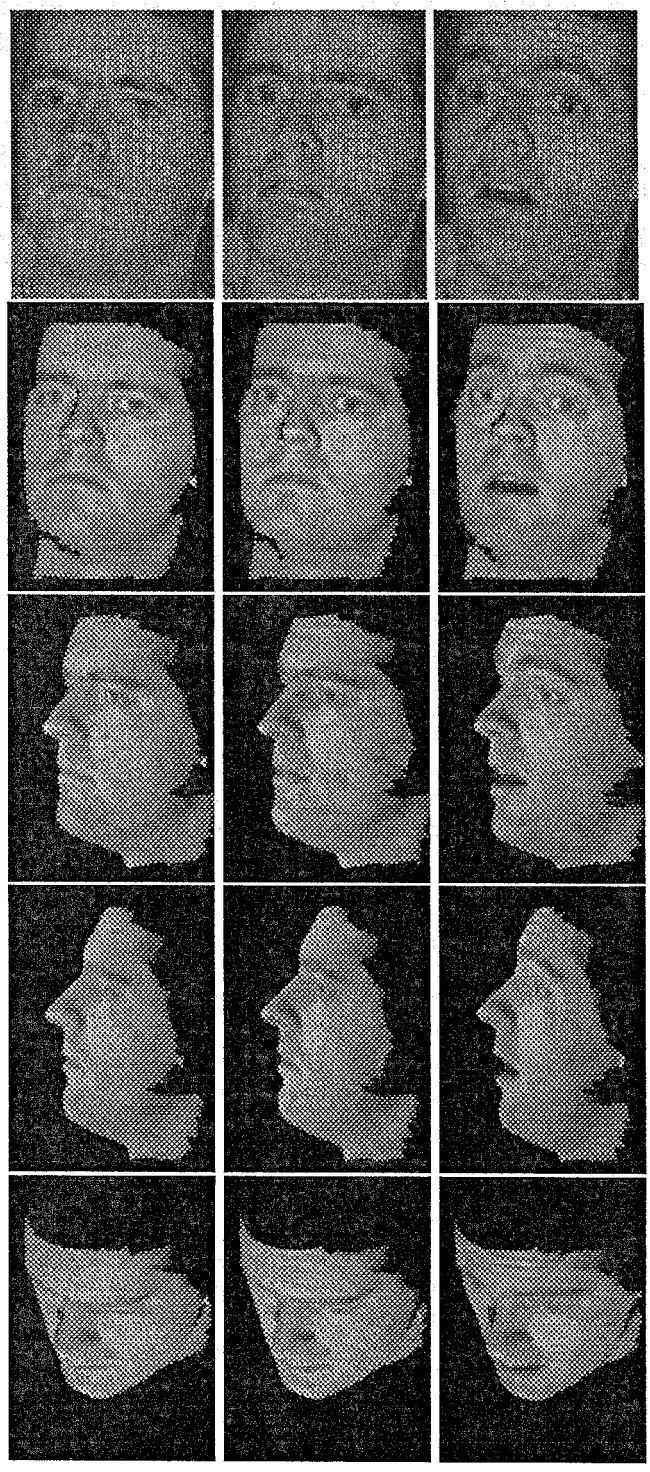

Figure 7: *Reconstruction of a talking face. Top row: input images with the projected pattern are frames from a video sequence; other rows: reconstructions seen from different view points.*