Optical Camera Tracking in Virtual Studios: Degenerate Cases

Athanasios Drosopoulos, Yiannis Xirouhakis and Anastasios Delopoulos
National Technical University of Athens
Image, Video and Mutlimedia Systems Laboratory
Zografou, 15773 Athens, Greece
{ndroso,jxiro}@image.ntua.gr

Abstract

Over the past few years, virtual studios applications have significantly attracted the attention of the entertainment industry. Optical tracking systems for virtual sets production have become particularly popular tending to substitute electro-mechanical ones. In this work, an existing optical tracking system [1] is revisited, in order to tackle with inherent degenerate cases; namely, reduction of the perspective projection model to the orthographic one and blurring of the blue screen. In this context, we propose a simple algorithm for 3D motion estimation under orthography using 3D-to-2D line correspondences. In addition, the watershed algorithm is employed for successful feature extraction in the presence of defocus or motion blur.

1. Introduction

The estimation of camera egomotion is a task of major interest in the fields of computer vision, pattern recognition and video understanding. Virtual studios applications have a lot to benefit from advances in these fields, regarding the topics of chromakeying, compositing and, in particular, 3D camera tracking (see [8] and references therein). Camera tracking systems employed in virtual studios are generally classified into two broad categories, namely the electro-mechanical and the optical ones. Several virtual studio systems have been developed, including Elset, 3DK, Synthevision for electro-mechanical and Cyberset, Mindset, the Mona Lisa Project for optical tracking among others. Electro-mechanical tracking is widely adopted since it is considered highly accurate, however it requires time-consuming pre-calibration procedures, sensors suffer from random vibrations, and the designated equipment can be very expensive. Optical tracking systems rely on pattern recognition schemes to extract camera motion on the basis of the frames captured. In turn, optical tracking fails when the referenced features are out of focus, occluded or even out of view. Moreover, markers cause compositing problems, in order to be made distinguishable from the blue background [8].

In [1], a method for the construction of a two-toned blue screen was introduced while an accompanying algorithm for 3D camera motion estimation from 3D-to-2D line correspondences was proposed in [12]. The features tracked in the particular system rely on the implicit line grid formed on the blue screen by its construction process. For the 3D camera motion estimation problem, several algorithms have been proposed in the literature. Using lines as input features and establishing correspondence has been also considered by [6, 3, 11] among others. For the same purpose, point correspondences are also considered for perspective and orthographic projection; see for example [10] and [9, 13]. In general, lines are preferable to points since they can be more accurately detected [1], however relatively many views and correspondences are required to solve for 3D motion. The utilization of 3D-to-2D line correspondences in [12] along with their placement on a rectangular grid reduced the required correspondences and views to four and two (one of which 2D) respectively; in the same context in [1], four point correspondences were used using a variant of [7], however lines were superior in terms of accuracy. In this way, the system overpasses nearly all problems of optical tracking. In addition, as an optical system itself does not suffer from time-costly pre-calibration and camera vibrations. However its performance is limited in degenerate cases; namely, in the presence of motion and defocus blur on the background (blue screen) or when the perspective projection model reduces to orthographic. The latter is the case when the camera is located far from the object.

In this work, both degenerate cases are addressed. A simple algorithm for 3D camera motion estimation on the basis of 2D-to-3D line correspondences under orthography is proposed, as an alternative to the algorithm given in [12] for the perspective case. At the same time, the watershed technique is applied in order to efficiently extract line features, even in the presence of significant defocus or motion

0-7695-0750-6/00 $10.00 © 2000 IEEE

blur, where traditional edge detection techniques perform poorely, and 3D camera motion parameters were estimated with particular accuracy.

2. Background and Notation

In [1], the screen is divided in rectangles, each one painted using one of two close levels of blue. Then the blue screen can be defined by a respective binary map $\mathbf{B}$ along with its real-world dimensions. In each captured frame, a small portion of the wall is within the camera field of view corresponding to a submatrix $\mathbf{S}$ of $\mathbf{B}$. Algebraic coding techniques (maximal length sequences) are employed in the construction of the binary map to ensure that any possible $\mathbf{S}$ exceeding a minimal size can be uniquely localized in $\mathbf{B}$. The camera field of view is determined with a strategy that uses no apparent feature correspondences for camera motion estimation, poses no problems to the system's chromakeying and compositing modules and estimates camera location rather than motion on the basis of the frame currently captured.

At the same time, a simple algorithm for the 3D camera motion estimation has been proposed for the particular scenario. The implicit rectangular grid formed by the rectangles' boundaries is modeled by two sets of 3D lines: a set of 'vertical' $X_v = \{X = x_i, i = 1 \cdots N\}$ and a set of 'horizontal' lines $Y_h = \{Y = y_j, j = 1 \cdots M\}$. The 3D lines along with the screen depth z_0 in the reference scene contain all geometric information required. The perspective pinhole camera model yields $x = f\frac{X}{Z}$ and $y = f\frac{Y}{Z}$, where (x, y) denote the cartesian coordinates of a point (X, Y, Z) projected onto the image plane. 3D camera motion is then obtained on the basis of the (virtual) reference scene and the current frame along the lines of [12]. In the latter, it is proved that 3D rotation, 3D translation and scale are efficiently determined even in the presence of noise. In short, from the currently captured frame, after chromakeying, 2D lines are extracted using Sobel filtering and the Hough transform. The cartesian line representation is employed for the extracted lines, $y' = a\,x' + b$, where (x', y') denote cartesian point coordinates in the current frame. The two sets of 3D lines project onto two corresponding sets of 2D lines in the current frame, L_v and L_h respectively. In this sense, let (a_{vi}, b_{vi}) and (a_{hj}, b_{hj}) be the line parameters of the i-th and j-th element of sets L_v and L_h respectively.

In order to accurately determine scale, the smallest possible rectangle is extracted (one such rectangle is assured to be contained in every frame by the blue screen construction method). Exact line correspondences *are not* required for the rotation estimation, whereas in order to determine translation, correspondence is established on the basis of the binary map.

3. Solving for 3D motion under orthography

In the orthographic case, projection model equations change to $x = X$, $y = Y$. In this case, it is clear that any 3D motion estimation algorithm for perspective projection is inapplicable, since the projection model becomes degenerate. However, as it will be shown, one can utilize the following algorithm as an alternative. By combining the line equation with the projection model eqs.,

$$\begin{bmatrix} a & -1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \end{bmatrix} + b = 0 \tag{1}$$

A horizontal line can be given in vector form in 3D space, as $[X\;Y\;Z]^T = [X\;y_j\;z_0]^T$, while its movement equation as

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \mathbf{R}_{2\times 3} \begin{bmatrix} X \\ y_j \\ z_0 \end{bmatrix} + \mathbf{t}\,, \tag{2}$$

where $\mathbf{R}_{2\times 3} = [r_{mn}] \overset{\triangle}{=} [\mathbf{r}_1\;\mathbf{r}_2\;\mathbf{r}_3]$ contains the first two rows of the rotation matrix $\mathbf{R}$, $\mathbf{r}_k$ is its k-th column and $\mathbf{t}$ is the 2D translation vector. The equation yielding the Z' coordinate has been ommitted, since it provides no additional information under orthography. In the same sense, the third component of $\mathbf{t}$ cannot be obtained.

In both sets of 2D lines for perfectly accurate measurements, line parameter a remains constant due to the nature of the projection model. In this sense, $a_{vi} \overset{\triangle}{=} a_v$ and $a_{hj} \overset{\triangle}{=} a_h$ for every i and j. Then, by combining equations (1) and (2), and since (2) holds for every X on the 3D line,

$$\begin{aligned} \begin{bmatrix} a_h & -1 \end{bmatrix} \mathbf{r}_1 &= 0\,, \qquad \text{and} \\ \begin{bmatrix} a_h & -1 \end{bmatrix} (y_j \mathbf{r}_2 + z_0 \mathbf{r}_3 + \mathbf{t}) &= -b_{hj}\,. \end{aligned} \tag{3}$$

Similarly, for a vertical line,

$$\begin{aligned} \begin{bmatrix} a_v & -1 \end{bmatrix} \mathbf{r}_2 &= 0\,, \qquad \text{and} \\ \begin{bmatrix} a_v & -1 \end{bmatrix} (x_i \mathbf{r}_1 + z_0 \mathbf{r}_3 + \mathbf{t}) &= -b_{vi}\,. \end{aligned} \tag{4}$$

After algebraic computations using eqs. (3) and (4),

$$r_{21} = a_h\, r_{11} \quad \text{and} \quad r_{22} = a_v\, r_{12}\,, \tag{5}$$

while for a pair of horizontal (y_1, y_2) and a pair of vertical (x_1, x_2) lines,

$$\begin{aligned} (y_2 - y_1)\,(a_h - a_v)\, r_{12} &= -(b_{h2} - b_{h1})\,, \\ (x_2 - x_1)\,(a_v - a_h)\, r_{11} &= -(b_{v2} - b_{v1})\,. \end{aligned} \tag{6}$$

In this sense, the rotation matrix $\mathbf{R}$ is estimated from eqs. (5) and (6), and the orthogonality equations, and since the depth of the screen z_0 is known, the 2D translation vector $\mathbf{t}$ is determined by eqs. (3) and (4).

It is interesting to notice, that by having determined a grid rectangle in the particular frame (for example distance $x_2 - x_1$ and $y_2 - y_1$), 'absence' of line features due to large contours of the same blue shade is detected by

$$y_3 - y_1 = (y_2 - y_1)\frac{b_{h3} - b_{h1}}{b_{h2} - b_{h1}}, \quad (7)$$

$$x_3 - x_1 = (x_2 - x_1)\frac{b_{v3} - b_{v1}}{b_{v2} - b_{v1}}, \quad (8)$$

for a third vertical (x_3) or horizontal (y_3) line detected. In fact, as it can be seen from eq. (6), exact line correspondences are not required for the estimation of $\mathbf{R}$; it suffices that for any two (a_v,b_{v1}) and (a_v,b_{v3}) corresponding to vertical lines x_1 and x_3, $x_3 - x_1$ is known, which in fact is ensured by eq. (7). On the contrary, establishment of line correspondences is required for the estimation of $\mathbf{t}$, which is made possible by the strategy adopted in the construction of the blue screen [1].

Since $a_{vi} \equiv a_v$ and $a_{hj} \equiv a_h$ under orthography, one can securely determine when to utilize the proposed approach instead of the one given in [12]. Before employing any of the two alternatives, $c_v = \sum_{i=1}^{N} a_{vi}$ and $c_h = \sum_{j=1}^{M} a_{hj}$ are computed. Thresholding the variance of a_{vi}s and a_{hj}s around c_v and c_h respectively, one can determine whether the perspective projection model reduces to the orthographic one or not. It must be finally pointed out that equations (6) are linear equations of one unknown and it is trivial that they are solved in the Least Squares sense for all 2D lines extracted in the current frame.

4. Extracting line features from blurred images

The efficiency of the proposed system relies, in a significant degree, upon the acurracy with which line features are extracted from the captured frame. It is expected that in a real-world system the line extraction algorithm should have a robust performance under any circumstances. Since this process is actually an edge detection algorithm followed by the Hough transform [1], it can be seen that incorrect measurements will affect directly the output of the former.

The standard edge detection approach fails to reliably localize edges in the captured video sequence, where blur from defocus and motion or even penumbral blur and shading can be a usual case. In this way, the blue screen lines project onto the image plane (frame) as a gradual luminance transition. It is widely approved that a blurred image is mathematically modeled by the convolution of a non-blurred image with a Gaussian blurring kernel [2, 5]. In general, a wide variety of techniques for edge detection from blurred images have been presented in the literature, a thorough review of which is beyond the scope of this work.

In order to successfully extract line features even in the presence of blur, present work employs the idea of the watershed for the edge detection task. The watershed technique, as described in [4] is a morphological filter, commonly used for segmentation purposes. In mathematical morphology, an image is modeled as a topological surface considering its intensity as the altitude. In this way, it is rather straightforward to estimate the variation from the gradient of the image. The watershed technique characterizes regions around local minima of the gradient as catchment basins, in the sense that if the image is gradually immersed into water, it is impossible for the water to reach another region of the image. When the water progressively floods the basins it will reach an altitude where two basins correspond. In this local maxima, a dam is raised to prevent the merging of different basins. Once the water reaches the global maxima, the set of dams raised constitutes the watershed of the image. Actually, in two dimensions, dams are raised along the curves that cross the boundary points between two basins. In this context, it can be seen that the watershed algorithm is most appropriate in our case, since image blur is by no means uniform, mainly due to the variation of background depth.

The result of the watershed technique is commonly an oversegmented image containing the correct set of contours. In the discussed system though, oversegmentation poses no considerable constraint, since the image contains no small segments, consisting of mainly two intensity values. An additional advantage is that the resulting edges are continuous, as the algorithm is designed for segmentation, and has single-pixel width even in the case where a plateau connects two basins. Experimental results have shown that even in frames with significant blur the extracted lines are almost identical to those extracted from the original image.

5. Simulation

A blue screen plane was constructed, along the lines of [1], in a virtual environment using an appropriate commercial software package. A virtual camera was then utilized to render blue screen's portions, for known camera motion parameters. The algorithm's performance under orthography was tested over a number of simulated experiments, yielding remarkably accurate camera motion estimates. Indicative results for the estimated rotation angle (dash-dotted line) versus its true values (solid line) are depicted in Figure 1 for a rendered sequence of 10 frames, where camera motion parameters were arbitrarily varied along time.

Rotation axis and angle were chosen indicatively for frame 7 $\mathbf{u} = [0.53\,0.80\,0.27]^T$ and $\alpha = 30^o$, whereas translation $\mathbf{T} = [-140\,50\,300]^T$. After the 2D lines were extracted from the captured frame, they were fed to the algorithm of Section 3 yielding $\hat{\mathbf{u}} = [0.52\ 0.77\ 0.26]^T$, $\hat{\alpha} = 29.8^o$, $\hat{\mathbf{t}} = [-140.1\ 49.8]^T$.

In Figure 2, significant non-uniform blur has been added to the captured frame. Nevertheless, the implementation of the watershed algorithm successfully extracted the correct edges (Figure 3). The estimated camera motion parameters did not again significantly vary from the true ones.

6. Conclusions

In this work, an existing optical tracking system [1, 12] for virtual sets production is revisited. In order to tackle common degenerate cases in real-world shooting, the particular system is extended to include a simple algorithm for 3D motion estimation under orthography and a watershed algorithm implementation for successful feature extraction in the presence of defocus or motion blur. The existing system along with the proposed extensions can be a powerful tool for commercial production of virtual sets.

References

[1] A. Drosopoulos, Y. Xirouhakis and A. Delopoulos. An Optical Camera Tracking System for Virtual Sets Applications. In *IEEE VMV Workshop*, Erlangen, Germany, Nov. 1999.

[2] A. Pentland. A new sense for depth of field. *IEEE Trans. PAMI*, 9(4):523–531, 1987.

[3] L. Quan and T. Kanade. Affine Structure from Line Correspondences with Uncalibrated Affine Cameras. *IEEE Trans. PAMI*, 19(8):834–845, 1997.

[4] L. Vincent and P. Soille. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Trans. PAMI*, 13(6):583–598, June 1991.

[5] M. Subbarao. Parallel depth recovery by changing camera parameters. In *ICCV*, pages 149–155, Tampa. Fla., 1988.

[6] M.E. Spetsakis and J. Aloimonos. Structure from Motion Using Line Correspondences. *Int'l J. Computer Vision*, 4:171–183, 1990.

[7] R.Y. Tsai, T.S. Huang and W.L. Zhu. Estimating 3-D motion parameters of a rigid planar patch II: Singular value decomposition. *IEEE Trans. Acoust. Speech Sign. Proc.*, 30(4):525–534, 1982.

[8] S. Gibbs et al. Virtual studios: an overview. *IEEE Multimedia*, 5(1):18–35, 1998.

[9] Tomasi, C., and T. Kanade. Shape and Motion from Image Streams under Orthography: a Factorization Method. *Int'l J. Computer Vision*, 9(2):137–154, 1992.

[10] Tsai, R.Y., and T.S. Huang. Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces. *IEEE Trans. PAMI*, 6:13–27, 1984.

[11] Y. Liu, T.S. Huang and O.D. Faugeras. Determination of camera location from 2-D to 3-D line and point correspondences. *IEEE Trans. PAMI*, 12(1):28–37, 1990.

[12] Y. Xirouhakis, A. Drosopoulos and A. Delopoulos. Camera Motion Estimation Using 3D-to-2D Line Correspondences. In *IEEE Nordic Signal Processing Symposium*, Kolmarden, Sweden, June 2000.

[13] Y. Xirouhakis and A. Delopoulos. Least Squares Estimation of 3D Shape and Motion of Rigid Objects from their Orthographic Projections. *To appear in IEEE Trans. PAMI*.

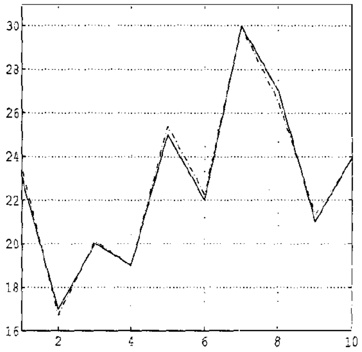

Figure 1. Real and estimated rotation angle values

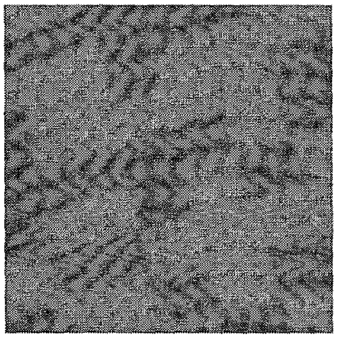

Figure 2. A blurred captured frame

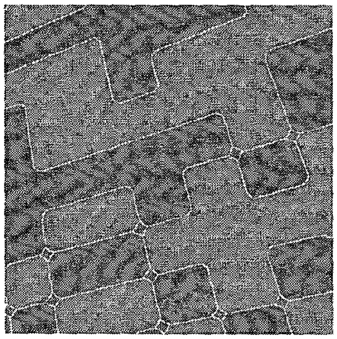

Figure 3. The watershed algorithm result