Self-calibrating stereo vision based 3D pointer device

Otto Korkalo
Helsinki University of Technology
otto.korkalo@tkk.fi

This paper presents a self-calibrating stereo vision system for human-computer interaction. Two low-cost off-the-shelf USB cameras are used to acquire stereo video, and based on the corresponding epipolar geometry a point of the three-space is tracked. The system allows us to use any point-like light source as 3D cursor. The internal parametrization of the cameras is obtained by self-calibration procedure, which requires only minimal actions from the user. As the internal parametrization is known, it is possible to recover the relative pose of the cameras and tracked point in either projective or euclidean space. Tracking is smoothed in image domain using standard Kalman filter to reduce noise.

1. Introduction

Direct three dimensional manipulation of objects is one of the major issues in human-computer interaction (HCI), when systems and user interfaces exploiting post-WIMP (windows, icons, menus and pointing devices) metaphors [1] are developed. In applications like virtual reality assisted CAD [10], the user is able to create and edit 3D content in truly 3D space. In the field of virtual reality, there has been long-term research considering different input devices, and many alternative technologies have been adopted to track the hands, head and position of the user. These include trackers based on e.g. magnetic, acoustic and optical signals, from which the methods relying on machine vision and photogrammetry have been gained great interest in the community lately.

Virtual reality systems and other installations that offer direct three dimensional interaction have been mostly utilized in research laboratories since the technology has been expensive and awkward to use. One of the drawbacks of many systems proposed, is the calibration procedure needed to getting started. Calibration patterns or other rigs are cumbersome to use, and if the aim is to develop easy-to-use systems for consumer markets, these subjects must be overcome. There are some commercial products which rely on cheap imaging sensors and computer vision, but the 3D information that could be obtained photogrammetrically is not put in use widespread.

In this paper, a self calibrating stereo vision based 3D tracker is described. The goal of this work is to present a framework of a construction which offers three degrees of freedom tracking capabilities without any geometrical a priori information about cameras or tracked object. The framework is verified by presenting an implementation of a system, which could be included in applications targeted to consumer markets. The benefit of the proposed framework is the self-calibrating nature. In our opinion, the only task the user should be expected to do, would be connect the cameras to the computer, and set them anyplace he wants.

2. Stereo system setup

A stereo vision based framework was developed to track points in three-space. The framework follows partly the work of [6] and it consists of the six main modules, which are described as follows

Point extractor The module is responsible to extract interest points from the source images. We use simple light emitting diode (LED) as input cursor, which is extracted from both the camera views by thresholding. Here we use LEDs to keep image analysis simple, but in general, the interest point could be any recognizable feature such as the user's finger tip. The number of the interest points is not restricted, and it should be noticed, that we do not use any a-priori information about the geometrical relationship between the tracked points to perform the self-calibration procedure.

Kalman filtering Interest points are Kalman filtered [9] to reduce noise and to smoothen the motion of the cursor. Kalman filtering is turned off when calculating the fundamental matrix, since it may cause inaccuracies when matching corresponding points.

Camera self-calibration A self calibration procedure is carried out to determine the internal parametrization of the cameras. As the system is started, the parametrization of the cameras is derived automatically, and the only maneuver the user is expected to perform, is to move tracked point trough the space in front of the cameras.

Reconstruction of the cameras Cameras are reconstructed in either projective or euclidean space. Both the reconstructions define the relative exterior orientations of the cameras, and they are obtained without any prior knowledge of the imaging geometry.

Triangulation The tracked points are triangulated by solving the linear set of equations using direct linear transform (DLT). The triangulation reconstructs the 3D trajectory of the tracked points in either projective or euclidean space.

Visualization The result of the tracking procedure is presented in OpenGL [12] window as a simple wire-frame cube moves trough the space.

3 Theory behind practise

3.1 Pinhole camera

Cameras are modeled using pinhole camera model, which is a linear mapping from the Euclidean 3-space to the 2-space of the image plane. If we consider a point $\mathbf{X}$ in the world coordinate frame, we may apply the mapping, and project the point to the point $\mathbf{x}$ of the image plane as

$$\mathbf{x} = \mathtt{P}\mathbf{X}, \tag{1}$$

where $\mathtt{P}$ is 3×4 sized camera matrix. The camera matrix describes the internal and exterior orientation of the camera, and can be decomposed as

$$\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I} \mid -\mathbf{C}]. \tag{2}$$

The exterior orientation defines the camera pose in the world coordinate frame, and can be retrieved from the equation 2, where $\mathbf{C}$ tells the camera center and $\mathtt{R}$ is a rotation matrix defining its orientation.

$\mathtt{K}$ is called the camera calibration matrix, and it specifies the internal orientation of the camera. The internal orientation defines the projection from the camera coordinate frame to the image plane, and it has five degrees of freedom. The most general form of the camera calibration matrix is

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where $\alpha_x = fm_x$ and $\alpha_y = fm_y$ represent the focal length in terms of pixel dimensions in both the directions, s is the skew parameter and (u_0, v_0) is the principal point of the camera. In practice, most of the CCD cameras has orthogonal pixels and the s parameter can be considered to be zero. Furthermore, we assume that the pixels of the camera are square and $\alpha_x = \alpha_y = f$. A line from the camera center is called the principal ray if it is perpendicular to the image plane. It meets the image plane at the principal point.

In real life cases, the mapping (1) is not linear, and corresponding world points, image points and optical center of the camera are not collinear. Especially cheap lenses tend to distort the image, and radial distortion is the most common type of inaccuracy. Throughout this paper, the cameras are modeled ideally, but practical solutions to improve the camera model in terms of self-calibration exists (see e.g. [7]).

3.2 Epipolar geometry

If we consider a point $\mathbf{X}$ in world coordinate frame imaged in two views, and the projected points are noted $\mathbf{x}$ and $\mathbf{x}'$, the back projected vectors starting from the camera centers $\mathbf{C}$ and $\mathbf{C}'$ going trough $\mathbf{x}$ and $\mathbf{x}'$ intersect at $\mathbf{X}$. The points and both the cameras lie on a common plane π, which is knows as the epipolar plane. The intersections of the image planes and the epipolar plane are corresponding epipolar lines $\mathbf{l}$ and $\mathbf{l}'$, and the line connecting the camera centers is called the baseline. It intersects the image planes at the epipoles.

If all the image points $\mathbf{x}$ and $\mathbf{x}'$ in stereo system satisfy the following equation

$$\mathbf{x}'^{\top}\mathtt{F}\mathbf{x} = 0, \tag{4}$$

then $\mathtt{F}$ is called the fundamental matrix of the stereo. A point on the first image is projected as a line to the second one, and the resulting line is the corresponding epipolar line $\mathbf{l}' = \mathtt{F}\mathbf{x}$. Similarly, a point on the second image has a projection on the first image as $\mathbf{l} = \mathtt{F}^{\top}\mathbf{x}'$. Fundamental matrix copes the whole epipolar geometry in a single algebraic representation, and it can be derived numerically in various ways. The method used here, is the normalized 8-point algorithm [8].

If the intrinsic parameterization of the camera is known, we can use normalized image coordinates and write $\hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x}$. Normalizing both the image points and writing

$$\hat{\mathbf{x}}'^{\top}\mathtt{E}\hat{\mathbf{x}} = 0 \tag{5}$$

we end up with the essential matrix, which is a specialized case of the fundamental matrix. Essential matrix describes the relative translation and orientation of the cameras up to scale, and it has five degrees of freedom.

The essential matrix can be written as

$$\mathtt{E} = [\mathbf{t}]_{\times}\mathtt{R}, \tag{6}$$

where $\mathbf{t}$ is the translation vector and $\mathtt{R}$ defines the orientation. The relation between the fundamental matrix and essential matrix is obviously as

$$\mathtt{E} = \mathtt{K}'^{\top}\mathtt{F}\mathtt{K}. \tag{7}$$

3.3 Projective reconstruction

Knowing the fundamental matrix allows us to obtain a projective reconstruction of the cameras and tracked point. A projective reconstruction is related to a metric one by non-singular matrix $\mathtt{H}$, which defines the homography between the two reconstructions:

$$\mathbf{X}_e = \mathtt{H}\mathbf{X}_p,\ \mathtt{P}_e = \mathtt{P}_p\mathtt{H}^{-1} \text{ and } \mathtt{P}'_e = \mathtt{P}'_p\mathtt{H}^{-1}. \quad (8)$$

To obtain a projective reconstruction, we can choose the camera matrices corrensponding to the fundamental matrix $\mathtt{F}$ as follows

$$\mathtt{P} = [\mathtt{I} \mid \mathbf{0}] \quad \text{and} \quad \mathtt{P}' = [[\mathbf{e}']_\times \mathtt{F} \mid \mathbf{e}']. \quad (9)$$

The reconstruction of the tracked point is then retrieved by triangulation.

A projective reconstruction can be easily upgraded to metric, if some additional information about the scene is given. The homography matrix $\mathtt{H}$ has 15 degrees of freedom, and if provided five or more known world points, from which four are not coplanar, the parameters can be calculated.

Projective reconstruction is not the goal in this paper, but it may be easily upgraded to get the metric reconstruction and even scale. This would be practical solution for HCI applications, but again, it requires more from the user.

3.4 Metric reconstruction

The metric reconstruction in uncalibrated case is the main goal of this paper. The problem could be solved directly from the projective reconstruction using extra information provided from the scene, or applying a stratified method, where the reconstruction is upgraded from the projective case trough affine to the final metric one. In this paper, we are interested in obtaining the metric reconstruction from the fundamental matrix, but we first apply a self-calibration procedure to determine the internal orientations of the cameras. Thus, we are able to calculate the essential matrix, and we are actually dealing with the calibrated case.

Once the essential matrix is known, the cameras may be retrieved by assuming that the camera matrix of the first camera is $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and the camera matrix of the second camera is one of the following

$$[\mathtt{U}\mathtt{W}\mathtt{V}^\top \mid +\mathbf{u}_3] \quad \text{or} \quad [\mathtt{U}\mathtt{W}\mathtt{V}^\top \mid -\mathbf{u}_3] \quad \text{or} \quad (10)$$

$$[\mathtt{U}\mathtt{W}^\top\mathtt{V}^\top \mid +\mathbf{u}_3] \quad \text{or} \quad [\mathtt{U}\mathtt{W}^\top\mathtt{V}^\top \mid -\mathbf{u}_3],$$

where the singular value decomposition (SVD) of $\mathtt{E}$ is $\mathtt{U}\,\mathrm{diag}(1,\,1,\,0)\,\mathtt{V}^\top$ and

$$\mathtt{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

One of the solutions presented above is physically right, and the other three are not. The right camera pair can be retrieved by studying the depth of the interest point from both the cameras. The case of the correct camera pair is the one, where the depth is positive in both views.

3.5 Camera self-calibration

Camera self-calibration is a procedure where the internal orientation of the camera is determined without any external information about the scene. Traditionally, the self-calibration is obtained using famous Kruppa's equations [4], which employs the use of absolute conics. The method applied here [5] is more simple, and it is based on the properties of the essential matrix.

The essential matrix has two identical singular values and the third one must be zero [3]. Since the essential matrix is depending on the internal parametrization of the cameras (eq. 7), we may write a cost function in terms of $\mathtt{K}$. In [5] the cost function to be minimized is formulated as

$$C(\mathtt{K}_i, i = 1, ..., n) = \sum_{ij}^{n} \frac{w_{ij}}{\sum_{kl}^{n} w_{kl}} \frac{{}^1\sigma_{ij} - {}^2\sigma_{ij}}{{}^2\sigma_{ij}}, \quad (12)$$

where ${}^1\sigma_{ij}$ and ${}^2\sigma_{ij}$ are the two non zero sigular values of $\mathtt{E}$ and ${}^1\sigma_{ij} > {}^2\sigma_{ij}$. In [5] the cost function was formulated for multiple pairs of point matches and fundamental matrices. As we derived only one $\mathtt{F}$, the weights w_{ij} can be ignored. Furthermore, we assumed the cameras to be identical, their principal points to the middle of the image, zero skew and square pixels, and so the only parameter to be determined is the common focal length f. Thus, the minimization problem is simplified and the solution can be obtained easily.

4 Results

The framework presented in this paper was applied, and a 3D tracker was implemented. The system consists of two identical low-cost off-the-shelf USB web cameras and software, which was written in C++ using Intel's OpenCV library [11]. The system runs in real-time, and it was tested in standard laptop using a single LED as input device. Image 1 shows the left view captured by the camera. Tracked point is highlighted, and the corresponding epiline is drawn over the image. The tracked point meets the epiline.

The path of the tracked point in image domain is presented in the images 2 a) and 2 b). The green polyline connects the original image points resulted from the point extractor module. The red line shows the Kalman filtered tracks, which are smoothed versions of the original ones.

Images 3 a) and 3 b) show the constructed tracks. In 3 a) a projective reconstruction is presented in a case, where

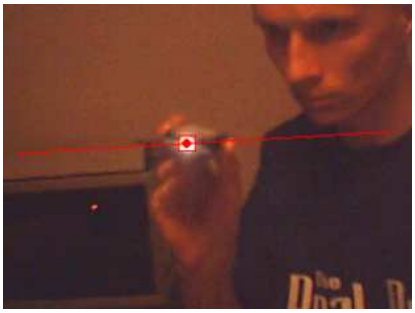

Figure 1: The left view of the stereo vision system. Tracked point meets the corresponding epipolar line.

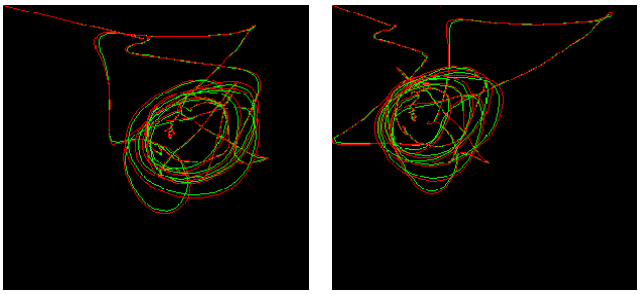

Figure 2: a) the left and b) the right image of the tracked point in image domain. Green lines show the original tracks, and the red ones the Kalman filtered paths.

the user is performing circle-like movement with the input cursor. Image 3 b) shows the metric reconstruction of the similar case.

The self-calibration procedure needs enough data to converge and obtain both the internal and exterior parameterization right. It was found out, that around 100 samples was enough data in our set-up. However, the system failed quite often to calibrate. This was mainly because there is delay between the left and the right image, and they are not synchronized. On the other hand, there seems to be some ill-posed imaging set-ups where the geometry causes the system to become numerically unstable. The geometrical analysis of the algorithms is needed in future work.

5 Discussion

A self-calibrating 3D pointer device for human-computer interaction was presented. The system enables us to use any point-like light source as input cursor in 3D, and it requires no a-priori information about the internal or exterior calibration of the cameras. We choose the system to rely on a single cursor, since tracking a point is more fundamental task compared to using input devices whose geometrical structure could be exploited in calibration tasks. The user is

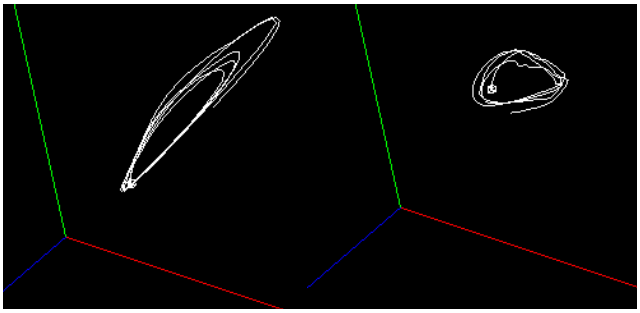

Figure 3: a) a projective and b) the metric reconstruction of the input cursor.

expected only to move single cursor in front of the cameras, and no calibration patterns, rigs or known-shaped input devices are needed. However the number of tracked points is not limited.

Implemented system verifies our approach. The system can successfully perform a self-calibration procedure, and it tracks the pointer in either projective or euclidean three-space. The proposed tracking module could be implemented in CHI systems targeted to consumer markets as it is simple to use and performs with various kind of imaging sensors.

Future work includes replacing the tracked point with more natural features, such as the fingertips of the user. The current system calibrates automatically as it is started, but it fails if the imaging geometry is changed. Calculating a cost function based on e.g epipolar error, the system could be made auto-recovering. The metric reconstruction is obtained here up to scale. In order to upgrade the system more applicable, the scale should be determined. This could be obtained by asking the user to move the cursor around the whole working space when the self-calibration procedure is carried out. A module for compensating the lens distortion effects will be implemented in future work, too.

References

[1] A. Dam. Post-WIMP user interfaces. *Communications of the ACM*, 40(2):63–67, 1997.

[2] R. Hartley and A. Zizzerman. *Multiple view geometry in computer vision* Cambridge University Press 2000

[3] T. Huang and O. Faugeras. Some prperties of the E-materix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1310–1312, 1989.

[4] S. maybank and O. faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[5] P. Mendonca and R. Cipolla. A simple technique for self-calibration. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (ICPR'99)*, pages 112–116, Fort Collins, Colorado, june 1999.

[6] D. Gorodnichy, S. Malik, and G. Roth. Affordable 3D face tracking using projective vision. In *Proceeding of Internation Confrence on Vision Interface (VI'02)*, pages 383–390, 2002.

[7] A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'01)*, pages 125–132, 2001.

[8] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133-135, 1981.

[9] R. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.

[10] M. Fiorentino, G. Monno and A. Uva. Smart tools for virtual reality based CAD. In *Proc. ADMAIAS04 International Confrence, Bari, Italy*, 2004.

[11] Intel's Open source computer vision library homepage. Cited 4/22/2006.
`www.intel.com/technology/computing/opencv/`

[12] OpenGL homepage. Cited 4/22/2006.
`www.opengl.org`