3D input from real-time object motion estimation for 3D interaction

Tserennadmid Tumurbaatar
Department of Computer and Information Science
National University of Mongolia, Ulaanbaatar, Mongolia
tserennadmid@seas.num.edu.mn

***Abstract*—the proposed algorithm is novel for the three-dimensional (3D) interaction system in real-time by determining 3D motion of a moving object in the image sequences taken from a single camera as its 3D input. The perspective model has formulated in non-linear least square problem to determine 3D motions as characterized by rotation and translation iteratively. In practice, it is numerically ill-conditioned for convergence of the equations, if it starts with not good enough initial guess. However, the results provided from para-perspective projection model are used as an initial value of non-linear optimization refinement under perspective camera model equations. The demonstration of 3D interaction procedures are developed in simple real-time 3D application system.**

Keywords—3D motion; perspective model; para-perspective model; 3D user interface

I. INTRODUCTION

3D interaction techniques and its 3D user interfaces have been a challenging task incorporating ability in machines for applications of video gaming, very large displays and mobile applications. In order to design usable 3D user interfaces, spatial tracking technologies of user's position, orientation, or motion are necessary to enable input to be used for 3D interaction. Most widely known 3D interface of console gaming systems uses dedicated sensor devices to track user's information for their own 3D input. For example, Wii Remote uses accelerometers to sense 3D motion and an infrared camera to sense pointing direction. Similarly, Move controller developed by Sony uses additional sensors to determine absolute 3D position and orientation of itself. Microsoft Kinect uses infrared lights and cameras to determine 3D position and pose of the entire body without a controller [1]. For most of interactive or animation applications, user interfaces are also performed indirectly with the 2D input devices by manipulating 2D widgets, entering coordinates, or choosing items from a menu [2]. Current commercial methods with dedicated devices and previous research works for spatial tracking offer methods and systems for 3D interaction in virtual environment applications. However, these still have limitations in terms of accuracy of motion estimation in depth dimension, the additional cost of dedicated devices and/or robustness for real-time applications. In this paper, we try to use a monocular web camera instead of other specialized devices since it is cheap and widely available. Furthermore, we believe hiring a single camera is more suitable for real-time computation as we can maintain one processing chain.

We will propose a new method for 3D interface through real-time 3D motion estimation of a moving object in image sequences taken by a single camera. For 3D motion estimation, we will utilize a non-linear minimization method that had been proposed for perspective model, and a para-perspective projection model that had been proposed as a solution for initial approximates. 3D motion estimated will be divided as 3D translation and 3D rotation and used as input for controlling a mouse and manipulating an object in a virtual 3D space in real-time. The contribution of this paper is that we propose a new method of 3D input by 3D motion estimation, and that users will be able to interact to 3D virtual world with 3D input from single camera in fast processing without requiring any additional hardware devices.

II. THE PROPOSED METHOD

A. Motion model

Consider that there are image sequences of the rigid object that is moving relative to a static camera as illustrated in Fig. 1.

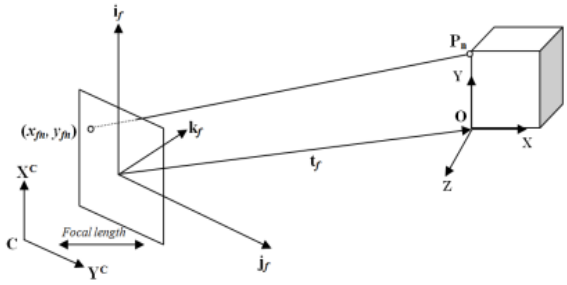


Fig. 1 The coordinate system of the camera and object

Each image is taken with keeping some object orientation that defined by the orthonormal unit vectors, $i_f = (i_{xf}, i_{yf}, i_{zf})$, $j_f = (j_{xf}, j_{yf}, j_{zf})$, and $k_f = (k_{xf}, k_{yf}, k_{zf})$ corresponding the x, y, and z axes of the

camera. We assume that N feature points are extracted in the first image, and are tracked to the next for each F image. N Feature points $P_n = (X_n, Y_n, Z_n)^T$ on the object that are projected into each F images with coordinates $p_{fn} = (x_{fn}, y_{fn}) | f = 1, \dots, F, n = 1, \dots, N$ under perspective projection. Initially, we formulate the equations based on rigidity constraint of the object. The 3D point in the object space is represented in camera coordinate system by a rotation matrix, R_f whose rows i_f, j_f, k_f and translation$t_f = (t_{xf}, t_{yf}, t_{zf})^T$.

$$P_n^c = R_f P_n + t_f \quad (1)$$

where

$$R_f = \begin{bmatrix} \cos\alpha \cos\beta & \sin\alpha \cos\beta & -\sin\beta \\ \cos\alpha \sin\beta \sin\gamma - \sin\alpha \cos\gamma & \sin\alpha \sin\beta \sin\gamma + \cos\alpha \cos\gamma & \cos\beta \sin\gamma \\ \cos\alpha \sin\beta \cos\gamma + \sin\alpha \sin\gamma & \sin\alpha \sin\beta \cos\gamma - \cos\alpha \sin\gamma & \cos\beta \cos\gamma \end{bmatrix} \quad (2)$$

The rotation matrix R_f is specified as three independent rotations around x, y and z axis by angles α, β, γ in equation (2).

Assuming the camera intrinsic parameters are known and focal length is unit throughout this letter, the relationship between the image space and the object space coordinates using the property of similar triangles can be written as:

$$x_{fn} = \frac{i_f \cdot P_n + t_{xf}}{k_f \cdot P_n + t_{zf}} \qquad y_{fn} = \frac{j_f \cdot P_n + t_{yf}}{k_f \cdot P_n + t_{zf}} \quad (3)$$

The 3D rotation and translation parameters can be obtained by formulating equation (3) in the non-linear least square problem with applying Taylor series first-order approximation. Since we have the given N point correspondences for each image F , a total of 2FN equations can be solved to determine six motion parameters ($t_{xf}, t_{yf}, t_{zf}, \alpha, \beta, \gamma$) and three shape parameters for each point ($P_n = [P_{n1}, P_{n2}, P_{n3}]$) for a total of $6F + 3N$.

B. Initialization through paraperspective approximation

Para-perspective projection closely approximates perspective projection by modeling image distortions of position effects and distance by developing para-perspective factorization method. The applicability of this factorization method is limited to offline computation for recovering shape and motion after all the input images are given, but we have used para-perspective factorization method in real-time to initialize non-linear system equations in (3). A more detailed description of the method can be found in [3]. Since we have the tracked N feature points over F frames in the image streams, we can write all these measurements into a single matrix to recover Euclidean shape and motion by SVD decomposition as follows:

$$\begin{bmatrix} x_{11} & \dots & x_{1N} \\ \dots & \dots & \dots \\ x_{F1} & \dots & x_{FN} \\ y_{11} & \dots & y_{1N} \\ \dots & \dots & \dots \\ y_{F1} & \dots & y_{FN} \end{bmatrix} = \begin{bmatrix} m_1 \\ \dots \\ m_F \\ n_1 \\ \dots \\ n_F \end{bmatrix} [P_1 \dots P_n] + \begin{bmatrix} t_{x1} \\ \dots \\ t_{xF} \\ t_{y1} \\ \dots \\ t_{yF} \end{bmatrix} [1 \dots 1] \quad (4)$$

In brief matrix form,

$$W = MS + T[1 \dots 1] \quad (5)$$

where $W \epsilon R^{2FxN}$ is the measurement matrix, $M \epsilon R^{2Fx3}$ is the motion matrix, $S \epsilon R^{3xN}$ is the shape matrix, and $T \epsilon R^{2Fx1}$ is the translation vector. Here,

$$t_{xf} = \frac{1}{N} \sum_{n=1}^{N} x_{fn} \qquad t_{yf} = \frac{1}{N} \sum_{n=1}^{N} y_{fn} \quad (6)$$

The decomposition (5) is not unique, but it is unique only up to an affine transformation. However, true motion and shape matrix can be determined by any Q invertible 3x3 matrix. In order to find Q, we can determine the metric constraints since the rows of the true motion matrix M are unit vectors:

$$\frac{\hat{m}_f^T Q Q^T \hat{m}_f}{1 + {t_{xf}}^2} - \frac{\hat{n}_f^T Q Q^T \hat{n}_f}{1 + {t_{yf}}^2} = 0 \quad (7)$$

$$\hat{m}_f^T Q Q^T \hat{n}_f - \frac{t_{xf} t_{yf}}{2} \left(\frac{\hat{m}_f^T Q Q^T \hat{m}_f}{1 + {t_{xf}}^2} + \frac{\hat{n}_f^T Q Q^T \hat{n}_f}{1 + {t_{yf}}^2} \right) = 0 \quad (8)$$

These are constraint equations (7) and (8) from which we obtain 2F non-linear equations in coefficients of Q, are homogenous, thus we need the additional constraint to avoid trivial null solution. We may choose:

$$\hat{m}_1^T Q Q^T \hat{m}_1 = 1 + {t_{x1}}^2 \quad (9)$$

By substituting $L = QQ^T$, equations (7), (8), and (9) become linear for six unknowns since L is the 3x3 symmetric positive matrix. In contrast, we finally obtained $2F + 1$ equations for six unknowns, so at least three frames are necessary to compute L matrix. Next, all unknown motion ($i_f^{(0)}, j_f^{(0)}, k_f^{(0)}, t_{xf}^{(0)}, t_{yf}^{(0)}, t_{zf}^{(0)}$) and shape parameters as initial guess values for the non-linear equations (3) can be found. These parameters are iteratively refined by using the equations in (3) under perspective projection.

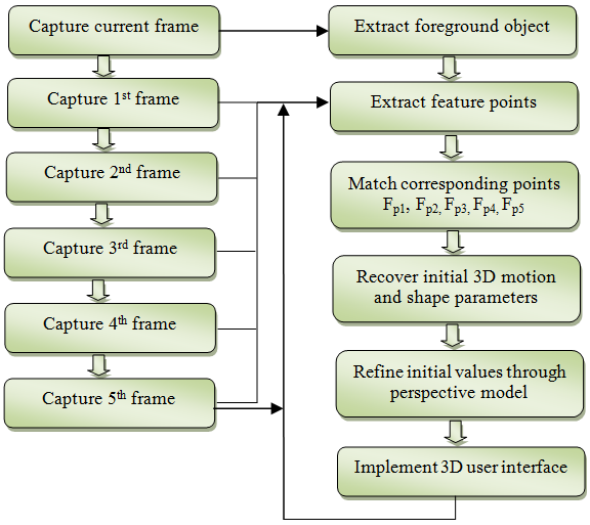


Fig. 2 The process flow diagram of the proposed method

III. EXPERIMENTS OF THE PROPOSED METHOD

A. Implementation of the proposed method

We implemented the system by using VC++, OpenCV, and OpenGL library on the 320x240 video sequences taken from a calibrated Microsoft LifeCam. Fig. 2 provides a process flow of the proposed method. We recovered motion parameters for every five frames. The first four frames are manually captured, and then 5th frame is captured consequently from current sequences. First, we extract a sub image, which is including only part of the foreground moving object, extracted from the current video sequence. Then, the feature points are computed to be matched to the next frames by using SURF or SIFT feature extractor for the extracted sub image. After these steps, the 1st, 2nd, 3rd, 4th, and the 5th frames are consequently captured, and the corresponding features are extracted between these frames and the extracted sub image. The best matches are found out by RANSAC (Random Sample Consensus) based robust method, eliminating outliers among the matched point with Brute Force matcher. Using the best matches, a perspective transformation (homography) matrix between these frames and the extracted sub image are found with the RANSAC-based robust method. The computed perspective transformation matrices for each frame are used to compute the exact corresponding points between the sub image and each captured frames since the input of the initialization step by para-perspective projection required the exact number of the tracked correspondences for all frames. To demonstrate the 3D interaction in the 3D virtual space, we performed application for drawing and controlling the 3D model represented by a blue as described in Fig. 3. The drawing actions of the 3D model are illustrated in the upper part of the window according to the estimated motion of the object or the camera. The object with computed feature correspondences $(F_{p1}, F_{p2}, F_{p3}, F_{p4}, F_{p5})$ in image sequences is shown in the bottom part of the window. Fig. 3 (a) is describing the object's rotation around z axis, Fig. 3 (b) is describing the object's rotation around y axis, and Fig. 3 (c) is describing the object's rotation around x axis clearly. The translation parameters along x and y axis are used as navigating mouse cursor into the their relative position of the 2D computer screen and navigating position of 3D model in 3D space by mapping screen coordinates of the mouse cursor into 3D window coordinate position.

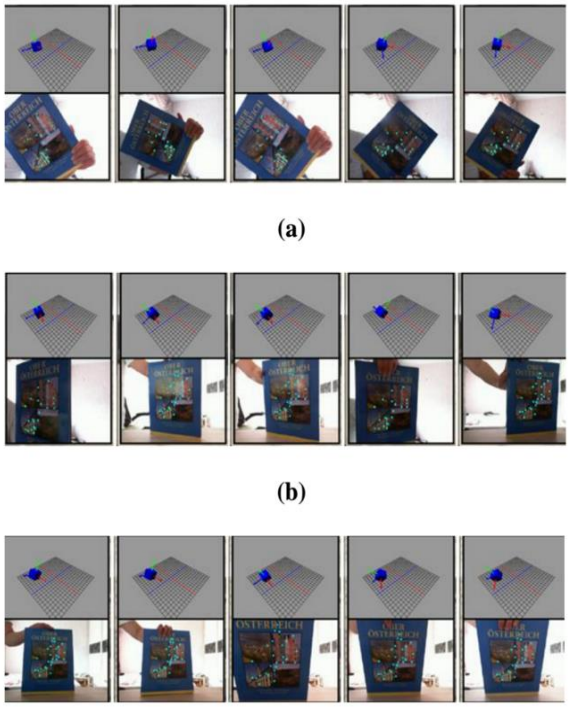
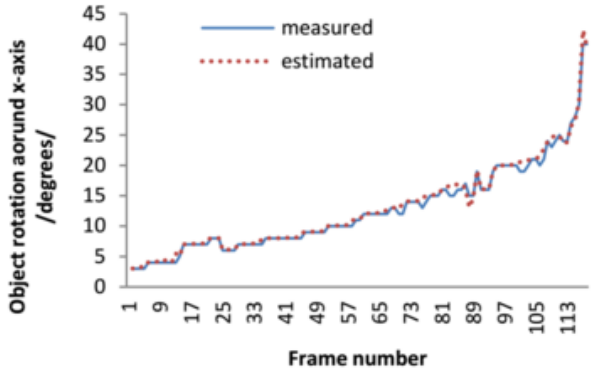
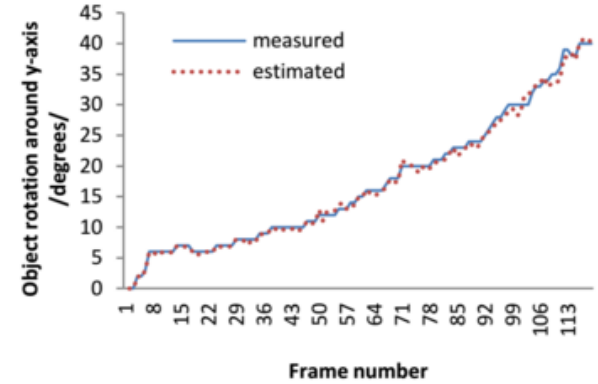


Fig. 3 Performance of the 3D user interface system with a moving object

B. Experiments

We examine the motion results by creating synthetic feature point sequences in a cube with known motion parameters. Each synthetic image was created by perspective projection with choosing largest focal length that keeps the object in the field of view throughout sequences. We rotate the synthetic cube through a total of 40 degrees for each axis. We checked the estimation accuracy of motion parameters through Root Mean Square (RMS) error of the estimated rotation parameters by comparing true rotation parameters for each frame. The synthetic dataset consist of 115 image frames. The estimated rotation parameters and the true rotation parameters are shown in Fig. 4. The computed RMS errors of rotation parameters around x axis, y axis and z axis were usually about 0.503 degrees, 0.61 degrees and 0.08 degrees, respectively. We extracted the feature points by both SURF and SIFT feature extractor. The events of the mouse click is handled by clockwise and anti-clockwise rotation around z axis for right and left button respectively. We set that the threshold value of angle is ± 15 for triggering click of the left and right button.





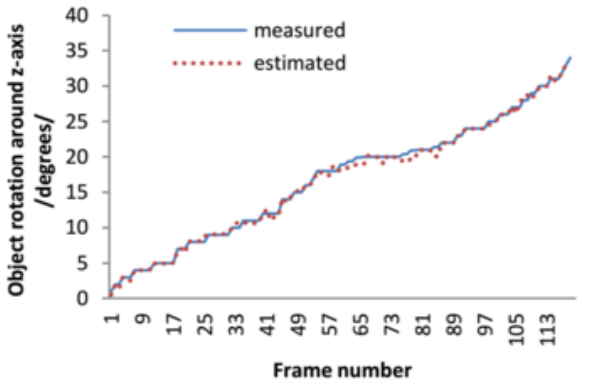


Fig. 4 Rotation results

CONCLUSION

We developed the 3D interaction system by estimating 3D motion parameters from rigid transformation equations with good initialization as its inputs when features in the 3D space and their perspective projections on the camera plane are known. One of the benefits in this study is every feature extractable object could be used to determine its 3D motion in our approach without requiring any additional hardware and model design. From the experiments, this approach is available for real time processing and useful for a number of 3D applications.

REFERENCES

[1] Doug Bowman, A.; Ryan Mcmahan, P.; Eric Ragan, D. Questioning Naturalism in 3D User Interfaces. Commun. ASM 2012, 55, 78-88

[2] Doug Bowman, A.; Coquillart, S.; Froehlich, B.; Hirose, M.; Kitamura, Y.; Kiyokawa, K.; Stuerzlingfer, W. 3D User interfaces: New directions and perspectives. IEEE Comput. Graph. Appl 2008, 28, 20-36

[3] Poelman, C.J., Kanade, T.,: 'A Paraperspective Factorization Method for Shape and Motion Recovery', IEEE Trans. Pattern Anal. Mach. Intell., March 1997, 19, (3)