An Innovative Geometric Pose Reconstruction Approach for Marker-based Single Camera Tracking

Pedro Santos
Fraunhofer IGD
Darmstadt, Germany
pedro.santos@igd.fhg.de

Alexandre Buaes, Carlos Eduardo Pereira
CETA SENAI-RS
PPGEE/UFRGS
Porto Alegre, Brazil
alexandre.greff@ufrgs.br, cpereira@ece.ufrgs.br

Alécio Binotto
CETA SENAI-RS
PPGC/UFRGS
Porto Alegre, Brazil
abinotto@inf.ufrgs.br

ABSTRACT

Mobile augmented reality applications are in need of tracking systems which can be wearable and do not cause high processing load, while still offering reasonable robustness, performance and accuracy. The motivation to develop yet another tracking algorithm is two-fold. Existing approaches use classical optimization techniques, which do not fully exploit the structure of the pose estimation problem with its geometric constraints. Also, mixed reality applications demand that pose estimation be not only accurate but also robust and computationally efficient. Hence there is a need for algorithms that are as accurate as classical algorithms, yet are also globally convergent and fast enough for real-time applications. In this paper we introduce a new iterative geometric method for pose estimation from four co-planar points and we present the current status of PTrack, an infrared marker-based single camera tracking system benefiting from this approach. Results show that our tracking system achieves competitive accuracy levels, while being highly stable and affordable.

Keywords: Tracking System, Augmented Reality, Virtual Reality.

1. INTRODUCTION

Immersive VR/AR application scenarios most often need continuous pose estimation information of objects to enable real-time interaction.

Optical tracking systems present accurate pose estimates at relatively high speed and low latency based on small user-worn components [Welch and Foxlin 2002]. The combination of infrared (IR) flash strobes and small retro-reflective markers still represents one of the most robust configurations against optical noise. This is why it is widely used by commercial systems for indoor applications. A marker-based approach also reduces required computational load for pre-processing tasks as opposed to markerless systems, allowing higher processing speeds. Because development of new hardware would be too expensive, use of standard cameras with PC-based processing was chosen.

The implementation of the new system consists of the hardware assembly (camera, IR flash strobes, connection to PC), implementation of a pre-processing software module, in order to extract the 2D position of markers on the camera’s sensor plane, and attachment of a tracking algorithm to the output of the pre-processing module in order to obtain 3D pose of tracked objects.

The presented work focuses on PTrack, an innovative tracking solution developed within two master theses [Santos 2005] [Buaes 2006], in cooperation projects among the authors’ institutions. The paper explains the implementation of the relevant algorithms and presents results of system performance tests.

2. RELATED WORK

Computing the position and orientation of a rigid body relative to the camera from a single view can be understood as the correspondence between a geometric feature in object space and its 2D projection in image space. This in turn can be expressed by a set of equations, whose unknowns are the extrinsic parameters. Generally, analytical and numerical methods can be distinguished. The first are represented by e.g. Fischler and Bolles [1981]. The goal of numerical approaches is to minimize an error function and thus iteratively approach the correct pose. One of the most prominent and accurate numerical approaches to pose estimation is the least square minimization from Lowe [1991]. ARToolKit [Kato and Billinghurst 1999] is also a numerical method. The drawbacks of numerical approaches can be conversion issues. However they are generally outweighed by less sensitivity to noise and a good speed vs. accuracy relation.

The presented single-camera system inserts itself into the numerical approaches. However, it takes advantage of geometrical conditions, namely that the corners of a square label in object space and the corresponding projection in image space must lie on the same projection lines commencing in the focal point and crossing the four corners of the label.

3. SYSTEM SETUP

A hardware module was specially adapted, consisting of an IDS uEye UI1210-C camera (Fig. 1), configured to acquire grayscale images up to 55 frames per second, equipped with IR flash strobes and a daylight blocking filter. The hardware module (without PC) costs around EUR 700.

In each frame the camera captures the image of labels and sends the uncompressed data to a PC through an USB 2.0 interface. The labels used consist of six markers each (Fig. 1). Four markers represent the corners of a square label with fixed edge length of 80 mm. One marker is on the top edge, identifying the top orientation of the label. The sixth marker is used to distinguish different labels and must lie somewhere within the square formed by the others. The camera recognizes markers from 60 cm distance from lens up to 3.5 m, resulting in 16 m³ working range.
Initially, a global thresholding algorithm is applied (to ensure more robustness against lighting variation), followed by a blob extraction algorithm (8-connect operator with region merging). Then both size and geometric constraints are considered in order to select only the regions with reasonable size (dimensions larger than 2x2 pixels) and shape (round or elliptical shape). Subsequently the center of each region is calculated with sub-pixel precision. Afterwards lens tangential and radial distortions are corrected and the intrinsic parameters of the camera are applied to the resulting coordinates (multiplication by camera calibration matrix). The results are the 2D real coordinates of the markers’ center on the plane of the camera’s sensor, which are then passed to pose estimation processing.

4.3. PTrackScan - 2D Pre-processing
Every new data-set with 2D marker center coordinates received from the Camera Module is processed by PTrackScan, where 2D pre-processing and 3D pose estimation take place.

4.3.1. Build and Scan Quad-tree
In 2D pre-processing the image space is divided using a quad-tree approach. The main goal of using a quad-tree structure is to benefit from the fact that all markers belonging to a label are always close by. If two labels are visible in an image space it is very likely that, using this approach, they will be in different quad-tree segments. Thus for recognition of a label in one segment no other markers belonging to another segment need to be considered, increasing processing speed.

For each node of the tree the algorithm checks if more than 5 unrecognized points are in this node. If this is not the case, then all points in the node are sent to the father node as unrecognized points. If more than 5 unrecognized points are found in the node, then the radar sweep algorithm continues processing. If a label is recognized it is added to the father-node's list of recognized labels. Unrecognized points of a node are always considered by upper nodes, so that label recognition is not jeopardized by the quad-tree scanning procedure. At the end, the root node contains a list of labels and still unrecognized points.

4.3.2. Radar-Sweep Algorithm
In case more than 5 unrecognized points are found in a node, then the possibility of detecting a label in that region exists. The radar-sweep identifies possible top-edges of potential labels containing three points on a line including the top marker. The main idea (Figure 3) is to use a radar-sweep line which will rotate up to 180 degrees around the node’s center of gravity. Once 3 collinear points have been found, the label detection algorithm takes over.

4.3.3. Label detection
When three points on a line are found and it is known if points above or below the line have to be considered for a potential label, then the algorithm tries to find the additional corners of the potential label. This is done by rotating a copy of the top edge around the first and the second corner which it connects, so both copies are perpendicular to the top-edge, connected to the respective corners of the top-edge. Then the projection of additional corners of a potential label must be close to the ends of both copies and can be found by calculating the distances of all points to be considered from those ends.
Once the nearest points are found and interpreted as additional corners, then a coding marker is found by searching for it between both additional corners. At the end of 2D pre-processing a list with potential labels is returned to PTrackScan as well as a list of unrecognized markers. Marker occlusion is not handled by the algorithm.

4.4. PTrackScan - 3D Pose Estimation

As the next step, 3D pose estimation for each potential label identified in the 2D image plane is attempted. If 3D reconstruction fails, or if no correspondence with a registered database label is established, then the potential label is not valid.

In order to obtain label orientation, let all 2D image plane coordinates of the potential label be translated to 3D camera space coordinates, what is possible due to knowledge of where the focal point (origin of the camera space) is in relation to the image plane in 3D. The projection lines commence in the focal point and cross each of the corners of the projected potential label. If reconstruction is possible, then corners of the reconstructed 3D label are on the same projection lines that cross the projection of those corners in the image plane. Therefore a mapping must exist between the projection and the original pose in 3D camera space.

Let one corner be visited at a given moment. If the projection of the edge parallel to its associated edge appears larger than the projection of the associated edge, then the label must be rotated counter-clockwise around the associated edge. If it appears smaller, then this rotation is clockwise. The difference of lengths between both parallel edges is compared before and after applying a rotation. In case the rotation causes the difference of lengths to increase, it is rolled back, and the algorithm proceeds to the next corner. The algorithm stops if the lengths of all edges are within a certain pre-defined accuracy limit.

In order to obtain the precise position in camera space of the label, scaling of the intermediary label is done along the projection lines until its edge length matches the standard edge length of all registered labels. After estimating the orientation and location of the projected potential label in camera space coordinates, a coordinate system transformation from camera space coordinates to object space coordinates of the label is applied. By analysis of the coding marker position, a direct comparison is done between the label and registered labels in the database, allowing its identification by its associated ID.

4.5. PTrackUDP

After computing the pose of all labels in camera space, the results are broadcasted to interested applications, using an output format which allows direct connection with Opentracker [Reitmayr and Schmalstieg 2001].

5. TESTS AND EVALUATION

Tests were conducted to infer system’s accuracy and precision for both translation and rotation situations. In the translation experiment a label is carried in uniform motion along a track. In the rotation test, the label is placed on a rotor with adjustable angle-of-attack mounted on a small motor. The angle-of-attack is defined as the angle at which the rotor is tilted from the position in which it is perpendicular to the axes of the motor.

In translation tests, Accuracy (average position error) reached 6.5 mm and Precision (standard deviation in position error), 4.6 mm. In rotation experiments, Precision and Accuracy values remained below 5 deg for angles-of-attack greater than 10 deg, and below 2.5 deg for angles-of-attack greater than 20 deg.

6. CONCLUSION

In this paper a new algorithm for iterative geometric pose estimation from four points was presented as well as an adequate 2D feature point pre-processing. The algorithms were embedded in a tracking system called PTrack. One can observe that the system allows for stable pose estimation results over time, which is important for augmented reality applications. In addition it is able to run at high update rates if needed and therefore provides small latency times for applications. For the future it is planned to extend the system to eventually support regular video cameras and provide large area tracking by using multiple cameras.

REFERENCES


