Tile Tracker: A Practical and Inexpensive Positioning System for Mobile AR Applications

Tile Tracker: A Practical and Inexpensive Positioning System for Mobile AR Applications

Steven Maesen, Yunjun Liu, Patrik Goorts, and Philippe Bekaert

Hasselt University - tUL - iMinds
Expertise Centre for Digital Media
Wetenschapspark 2
3590 Diepenbeek, Belgium
{steven.maesen,yunjun.liu,patrik.goorts,philippe.bekaert}@uhasselt.be

Abstract. In this paper, we propose a practical and low-cost positioning system that quickly determines the camera pose in an environment with pre-existing ground tiles, with the support of fiducial images. Such environment exists in many places, both indoors and outdoors, such as museums and parks. Our system is designed to be used on mobile devices (e.g. smartphone, tablets, etc.) with inexpensive cameras to fast calibrate cameras and to track the movement of a user. Contrary to other existing mobile tracking systems, our algorithm has minimal drift over time. To accomplish this, our approach, fully taking advantage of the existing rectangular tiles, labels them with markers that supports quick orientation determination and unique identification. Our algorithm recovers the camera pose based on the images taken of these marked tiles in real time, and thus provides basis for a wide range of applications such as navigational assistance and augmented or virtual reality.

Keywords: optical tracking, video tracking, mobile AR, marker, fiducial, low-cost

Fig. 1. Examples of ground tiles in natural environment. Rectangular grid pavement outside Louvre Museum (left) [14], floor tiles inside exhibition space (middle) [14] and stone tiles around the Lincoln Memorial Reflecting Pool (right) [2]

1 Introduction

Tracking is the process of locating moving objects or camera pose in consecutive video frames and is a field that has been long studied. It is a fundamental component of a variety of computer vision applications such as traffic monitoring control, security and surveillance, medical imaging, activity recognition and augmented reality. A typical strategy to solve the problem of object tracking in the 3D real world is to effectively extract features in each frame and reliably match corresponding features over time. Robust feature detection and matching algorithms, including SIFT [9], SURF [5] and FLANN [12], can be used for this purpose. However, since our goal is real-time tracking on mobile devices, these algorithms are computationally costly and impractical to implement on such devices with limited memory and processing power. Another widely-used approach is motion-based tracking algorithms that subtract the background from the moving object and determine the matching in subsequent frames using location prediction (e.g. Kalman filter). The object tracking methods proposed by Han et al. [8] and Aggarwal et al. [4] follow this approach. These methods also require considerable amount of image processing. Instead of tracking complex objects in the scene, we transform the problem into a planar object tracking problem using markers. Existing markers include square shaped ARToolKit [1] and QR code, etc., circular ones proposed by Ababsa et al. [3] or line code using De Bruijn sequence proposed by Maesen et al. [10] In our approach, we propose to use existing ground tiles to create an optical tracking system with minimal drift. To get a global camera pose, we incorporate a simple dataset of unambiguous and easily-identiable 2D rectangular markers to be used on the existing ground grid. Due to the simplicity and fast speed, our approach has real-time performance on mobile devices. One significant advantage of our approach is its cost-effectiveness because we fully exploit the existing ground tiles to reduce marker tagging and alignment cost, besides their quality features that serve to be part of the markers.

2 Overview of Our Approach

We propose a low-cost and practical optical tracking system that can be used with low resolution cameras (e.g. smartphone or tablets cameras) in an environment with rectangular ground grid (e.g. ceramic tiles). The tiles in the grid are tagged with uniquely identifiable markers and are recognized in real-time to determine the camera pose. Our system works as follows (see Figure 2): First, a series of images or video sequences (we call them frames henceforth) are taken as the mobile device user walks through the space. Our system processes each frame, using computer vision techniques to detect the tiles together with markers on them (Section 4). Then we match the feature points on the processed frame with our data-set of marker patterns and find the corresponding marker (Section 3). Finally, from the matching result and the known world coordinates of the real tiles, we recover the camera parameters (Section 5).

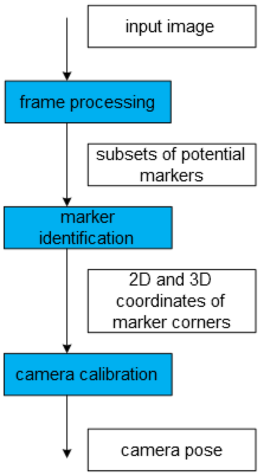

Fig. 2. System overview

3 Marking

Markers, also known as fiducial images, are a widely-used supporting element in vision-based tracking systems. The purpose of introducing markers into an environment for tracking is to support fast and deterministic identification and thus reliable localization. Once identification of the markers is accomplished, a homography can be established and the camera pose can be recovered from the image-world-correspondence.

3.1 Why Ground Tiles?

While markers enable fast recognition and tracking, they are not a free cake to take and a non-trivial cost comes with it. Besides a relatively insignificant raw material and printing cost, the primary cost involves a huge amount of human labour regarding tagging and aligning the markers in the physical environment in which object tracking is intended. This is tedious work and, depending on the scale of the environment, usually requires hours or days of work for more than one person, even with the assistance of optical measurement tools and alignment technology.

To reduce the amount of work, we propose a significantly faster marking approach that uses existing rectangular ground grid, e.g. ceramic floor tiles or square stone walkways, etc. We have observed that these grids exist in many places, such as supermarkets, museums, conference centres and parks (Figure 1). Many of these environments have the potential for AR or navigational applications. For example, a museum AR application can use visitors' tracked positions within the building to guide them through the artworks or exhibits, highlighting the pieces that might be of specific interests. A navigational application of a theme park can also pin-point the visitors and use this information to enhance their experience. As existing grids are already aligned, the labour of tagging in such environment is greatly reduced and becomes naturally effortless. The measuring of world coordinates of each tile is trivial as they can be calculated straightforwardly by adding an offset to an appointed origin in the grid. In our approach, we use rectangular tiles because they are simple, have clear computational advantages and yield great accuracy, potentially to sub-pixel level. However, the tiles are not required to be rectangular and any tiles that can emit four-point correspondences suffice.

3.2 Marker Design

Many marker designs exist and most commonly seen square markers are ARToolKit [1], ARTag [7], Tricodes [11], DataMatrix [15] and QR code [6]. Circular markers by Ababsa et al. [3] and line encoded markers by Maesen et al. [10] are also used for real-time tracking. The choice of the markers is directly related to the matching algorithm, therefore it is important to have good designs. There are certain criteria regarding how to choose good markers. As described by C.B.Owen et al. [13], the general guidelines are as follows:

- support unambiguous determination of position and orientation
- be easy to locate and identify using fast and simple algorithms
- members of the marker dataset should be unique and unlikely be confused with each other (i.e. minimal inter-marker correlation)

According to the guidelines mentioned above, we designed the marker dataset to use the following pattern (to be experimented in our approach): The existing ground grid serves as the border of the individual markers. (Please note markers are not limited to fit inside one tile. They can also be bigger and consist more than one tile.) A very distinctive advantage of rectangular marker is that it can yield four corner points for tracking purpose. The straight tile edges allow corner extraction to be less sensitive to noise in the vicinity or quantization errors. In our experiment, we used a black-white binary 4-by-4 circular inner pattern, as shown in Figure 3. The top-left dot is always white and the other three dots in the outermost corners (namely, top-right, bottom-left and bottom-right) are always black, and they are used to support the determination of unique orientation. The other dots can be either black or white.

Our marker design is very compact and satisfies the first two criteria. However, it doesn't have zero inter-marker correlation, but for our initial testing, it suffices. When we have to scale up in later stage of our project, or in an environment with high amount of noises, we can easily improve our marker pattern by adding a few bits of checksum for error-correcting, or using orthogonal encoding whose cross correlation is low.

3.3 Determining the Match

After we got the subsets of corner points from processing the frame, we extract the corresponding region enclosed by each subset of corner points in the binary

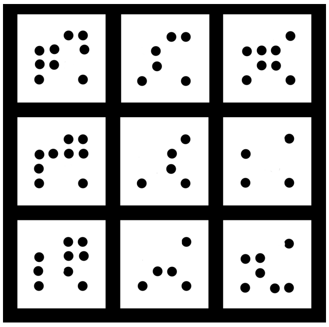

Fig. 3. Example of sample markers in our dataset

image we get from adaptive thresholding. This region can be warped to a common frame of reference (for example 4-by-4) for the identification of the marker in our dataset. (Figure 4)

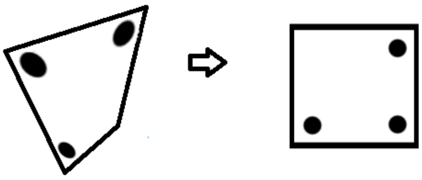

Fig. 4. The extracted image has to be warped and aligned before it can be compared with sample markers in the dataset.

After the orientation of the warped image is determined by the three outermost black dots, we rotate the warped image so that it is aligned with the sample markers stored in the dataset . As dots are in fixed positions, we can read them out in a fixed order (based on orientation). The black dots signal the binary value 0, while white gives the value 1. Reading them out gives us an unique identifier ranging from 0 to 2^{n-4}, where n equals the total number of bits (16 in our experiment). This speeds up the identification in a hashed database compared to image matching in other marker based systems.

4 Frame Processing

Before the markers can be successfully identified, we need to detect the tile floor. The only assumption we make about the used mobile camera, is the fact that is has a fixed or manual zoom which is usually the case. Therefore we can calibrate the intrinsic camera parameters first using an OpenCV checkerboard pattern. After this pre-processing step, the user can take images at any arbitrary angle or position when he walks through the marked space as long as some ground tiles are present in the image. When the video sequences or images are taken of the tiles, they are first processed in the following a few steps to achieve a higher quality of recognition.

4.1 Binarization

First, a binary image is created through thresholding on the gray-scale image of the original frame. Simple thresholding straightforwardly separates the foreground from the background using a single global value as its threshold. This method does not give a satisfactory binarization in our experiment because it does not deal with varying illumination or noise (e.g. shadows from surrounding objects cast on the ground tiles). In our approach we use adaptive thresholding which gives an improved binarization that responds better to local features under non-uniform lighting.

4.2 Segmentation and Contour Detection

Because our goal is to determine the camera pose by correctly matching the four corners of corresponding tiles, we first have to identify the tile corners. Given all tiles have distinct straight-line borders, a fast way to find the tile corners, is to treat each tile as a separate segment and extract the vertices on its contour as its corners. With the binary image that we've got from the previous step, we look for all the segments using connected components labelling algorithm and extract contours from the result segments. The points are stored as subsets (one subset per contour). As an optimization, we only store the end points (i.e. x, y values) of horizontal, vertical and diagonal segments, instead of all contour points. This significantly reduces requirement on memory space which is highly limited on mobile devices. Fewer stored points also mean less data to process thus an increase of the processing speed and this makes real-time response possible even when most mobile devices do not yet have high computing power as compared to desktops.

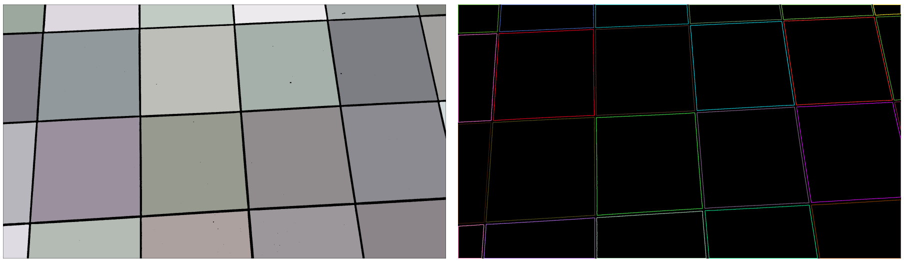

Fig. 5. Tiles are detected as segments using connected components labeling algorithm (left) and contours are extracted (right).

4.3 Corner Extraction

As we are only interested in points that are tile corners, now that we have subsets of contour points detected, we have to remove those that do not belong to the tiles. The idea is to use Ramer-Douglas-Peucker algorithm to simplify all the detected contours with fewer vertices. We choose an epsilon to preserve the approximate shapes of the contour and yet allow us to filter out the obvious non-rectangular-shaped outliers. To quickly remove non-tile points, we remove the contours that are geometrically not convex or consist of more than four vertices, because they cannot form a valid tile in the image. We notice that this step sifts out a big portion of outlier points that can otherwise be considered as valid tiles.

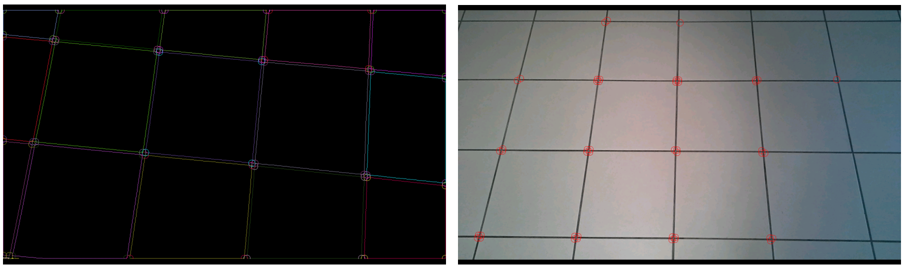

Fig. 6. Corners are extracted from contours (left) and marked on the input image (right).

5 Camera Calibration

Determination of the camera pose relative to an environment requires the correspondence of at least four identified 2D points in the camera image and their 3D world coordinates. If we incorporate the marker identification from Section 3, we can get the 3D world coordinates of each corner point. Because all points lay on a flat surface (the floor), a homography can be calculated which gives us a linear mapping between the camera plane (2D image points of the corners) and the floor plane. Using the standard pinhole model and the knowledge of the intrinsic camera parameters, we can estimate the rotation and translation of the camera and the mobile device. This can be used in any VR or AR application to generate the correct viewpoint.

6 Discussion

The work presented in this paper is still work-in-progress. Our first prototype works under controlled lab conditions on a Samsung Galaxy Note 10.1 tablet with a 1.4 GHz processor. Our first results look promising to further investigate the possible applications. Further testing in real world conditions is planned and extensive (latency) testing is needed on different mobile devices.

7 Conclusion

We presented a low-cost and practical optical tracking system that can be used in an existing environment with any rectangular ground grid. We proposed a simple and robust design of markers to be used with our system that supports quick orientation determination and unique identification. With the use of ground tiles, our system has very low cost and can be practically implemented with ease. Computationally, our algorithm is fast enough to be used on mobile devices with low resolution cameras. In the future we would like to incorporate inertial sensors for improved robustness. This also gives the opportunity of reducing the number of tiles that need markers to about one per square meter. We would also like to further optimize our algorithms to be more robust and faster.

References

1. http://www.hitl.washington.edu/artoolkit/
2. (Jul 2005), http://en.wikipedia.org/wiki/Lincoln_Memorial_Reflecting_Pool#mediaviewer/File:Reflecting_pool.jpg
3. Ababsa, F., Mallem, M.: A robust circular fiducial detection technique and real-time 3d camera tracking. Journal of Multimedia 3(4), 34–41 (2008)
4. Aggarwal, A., Biswas, S., Singh, S., Sural, S., Majumdar, A.: Object tracking using background subtraction and motion estimation in mpeg videos. In: Narayanan, P., Nayar, S., Shum, H.Y. (eds.) Computer Vision ACCV 2006, Lecture Notes in Computer Science, vol. 3852, pp. 121–130. Springer Berlin Heidelberg (2006), http://dx.doi.org/10.1007/11612704_13
5. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Computer Vision and Image Understanding 110(3), 346–359 (Jun 2008), http://dx.doi.org/10.1016/j.cviu.2007.09.014
6. Denso: Qr code essentials (March 2013), http://www.nacs.org/LinkClick.aspx?fileticket=D1FpVAvvJuo%3D&tabid=1426&mid=4802
7. Fiala, M.: Artag revision 1, a fiducial marker system using digital techniques. Tech. rep., National Research Council (Nov 2004)
8. Han, M., Sethi, A., Hua, W., Gong, Y.: A detection-based multiple object tracking method. In: Image Processing, 2004. ICIP '04. 2004 International Conference on. vol. 5, pp. 3065–3068 Vol. 5 (Oct 2004)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (Nov 2004)
10. Maesen, S., Goorts, P., Bekaert, P.: Scalable optical tracking for navigating large virtual environments using spatially encoded markers. In: Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology. pp. 101–110. VRST '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2503713.2503733
11. Mooser, J., You, S., Neumann, U.: Tricodes: A barcode-like fiducial design for augmented reality media. In: Multimedia and Expo, 2006 IEEE International Conference on. pp. 1301–1304 (July 2006)
12. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: VISAPP International Conference on Computer Vision Theory and Applications. pp. 331–340 (2009)
13. Owen, C., Xiao, F., Middlin, P.: What is the best fiducial? In: Augmented Reality Toolkit, The First IEEE International Workshop. pp. 8–16 (2002)
14. Ruault, P.: (2012), http://www.businessinsider.com/the-louvre-opens-islamic-art-wing-2012-9?op=1
15. Stevenson, R.: Laser marking matrix codes on pcbs (Dec 2005), http://www.thefreelibrary.com/Laser%20marking%20matrix%20codes%20on%20PCBS:%20the%20one-dimensional%20barcodes%20used...-a0140015287